AFRL-SR-AR-TR-09-0252

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)* 23-07-2009	**2. REPORT TYPE** Final Technical	**3. DATES COVERED** *(From - To)* 24 MAY 2007 - 31 DEC 2008

4. TITLE AND SUBTITLE Game-theoretic High-Level Distributed Fusion	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER FA9550-07-1-0474
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Chin	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) BAE Systems National Security Solutions Inc BAE Systems Advanced Information Technologies 6 New England Executive Park Burlington, MA 01803-5012	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 875 N. Randolph St Arlington, VA 22205	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFOSR
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Our research is mainly motivated first by our desire to develop a theoretical foundation for hierarchical dynamic game framework and secondly by the desire to demonstrate the applicability of such theory in the area of for sensor resource management of large sensor networks of highly decentralized, distributed nature for the purpose of multi-level data fusion. We briefly describe the historical context of game theory over the past 80 years out of which our research ideas were engendered. We then present three new enabling approaches to game theory that makes it more applicable than past applications of 2-person static game, and describe them in detail, including results of simulations.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU		

Final Report

Game-theoretic High-Level Distributed Fusion

AFOSR Research Grant

December 31, 2008

Submitted To:

Khanh D. Pham, Ph.D., DR-II

Decision Support Systems & Autonomy

Air Force Research Laboratory

Space Vehicles Directorate

AFRL/RVSV

3550 Aberdeen Ave. SE

Kirtland AFB, NM 87117

(505)-846-4823 (Voice)

(505)-846-7877 (Fax)

Submitted By:
Dr. Sang Chin

Division Chief Technology Officer
Science and Technology Division
Science Applications International Corporations
4001 No. Fairfax Drive, Arlington, VA 22203

ABSTRACT

Our research is mainly motivated first by our desire to devclop a theoretical foundation for hicr-archical dynamic game framework and secondly by the desire to demonstrate the applicability of such theory in the area of for sensor resource management of large sensor networks of highly decentralized, distributed nature for the purpose of multi-level data fusion. We briefly describe the historical context of game theory over the past 80 ycars out of which our research ideas were engendered. We then present three new enabling approaches to game theory that makes it more applicable than past applications of 2-person static game, and dcscribe them in detail, including results of simulations.

1. INTRODUCTION

1.1. BRIEF HISTORY

Game theory, a mathematical study of strategies in games, has a long and rich history dating all back to early 1700's, and found a firm mathematical footing in 1928 when von Neumann published his famous paper. In his work, Von Neumann showed that every two-person zero-sum game has a maxi-min solution in either pure or mixed strategies. In other words, in games in which one player's winnings equal the other player's losses, Neumann and Morgenstern showed that it is rational for each player to choose the strategy that maximizes his minimum payoff, giving rise to a notion of *equilibrium* solution, i.e. a pair of strategies, one for each player, with which each player can be most satisfied with, given that the other play does not alter his strategy.

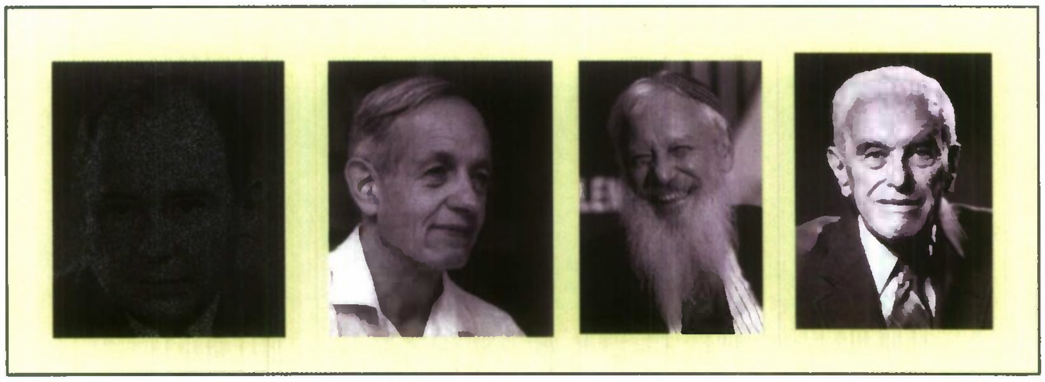


Figure 1: Von Neumann, Nash, Aumann, & Harsanyi – pioneers of 2-person, N-person, Dynamic and Uncertain Games

This seminal work brought to the ficld of game theory quite an excitement and even daring hope among some mathematicians that it would do to the field of economics what Newtonian calculus did for physics. Most importantly, it ushered in to the field two bright young researchers who would significantly extend the outer limits of game theory. A few such young minds particularly stand out. One is *John F. Nash*, a Princeton mathematician who, in his 1951 Ph.D. thesis, [give reference here] extended Neumann's theory to N-person non-cooperative games and established the notion of what is now famously known as Nash-equilibrium that eventually brought him the Nobel recognition in 1994. Another is *Robert Aumann*, who took a departure from this fast developing field of finite *static* games in the tradition of Neumann and Nash, and instead studied the *dynamic* version of games, where the strategies of players and sometimes games themselves change along a time-parameter *t* according to some set of dynamic equations that govern such a change. Aumann contributed much to what we know as theory of dynamic games today and his work eventually also garnered him a Nobel Prize in 2002. *Rufus Isaacs*, who took a departure from this fast developing field of finite discrete-time games in the tradition of Neumann, Nash, and Aumann, instead studied the continuous-time version of games, where the way in which strategies of players determine the state (or trajectories) of players depends continuously on time-parameter t according to some partial differential equation. Isaacs, who worked as engineer on warplane propellers during World War II, joined the mathematics department of RAND corporation, a cold war think tank after the war, and there he passionately developed the theory of such game-theoretic differential equations, a theory now called differential games, with the topic of warfare strategy as the foremost application in his mind. Around the same time, *John Harsanyi* made a key recognition of the imperfectness and incompleteness of information that is inherent in many practical games, and thus started a theory of uncertain games (often also known as games with incomplete information) and was also awarded a Nobel prize in 1994 along with John F. Nash.

1.2. CURRENT RELEVANCE AND NEED FOR NEW APPROACHES

Despite the preponderance of their military applications, however, the two-person games, research of which flourished under the clouds of the two super-power-cold-war, have limitations in this post cold war, post September 11 era. One obvious limitation is the apparent lack of an efficient 2-person game-solver. Of course, Neumann and Nash showed that such a solution exists and later in 1980's, it was shown that every finite 2-person game can be turned into a linear programming problem, which then can be solved using a number of different techniques, say, by a simplex method for example. However, such a translation is not only computationally cumbersome, but furthermore, some structure of the game, such as symmetry, asymmetry, dominance, etc., and thus some potential insight into the game are often lost during this translation, making it desirable for an efficient 2-person game-solver that preserves and even exploits structures of the game in question. *Our research is in part motivated by our desire to look for just such an efficient 2-person game solver.*

Moreover, game theory research applied to defense-related problems has heretofore mostly focused on static games. This was consistent with the traditional belief that our adversary has a pre-defined set of military strategies that he has perfected over many years (especially during the cold-war) as well as relatively short time of engagement during which one fixed adversarial strategy is committed. However in this post September-11 era, there is an increasing awareness that the adversary is constantly changing his attack strategies, and such variability of adversarial strategies and even of the game from which these strategies are derived call for application of dynamic games to address the current. However, although solutions of any dynamic 2-person game are known to exist by the Folk theorem, apparent lack of an on-line technique to solver dynamic games in an iterative fashion has stymied such an application. *In our research, we present just such an on-line iterative method to solve dynamic games using insights from Kalman filtering techniques.*

Furthermore, while the interaction between offense and defense can be effectively modeled as a 2-person game, the resource allocation problem among different assets of a sensor system can further be modeled as an N-person game. Thanks to Nash's seminal work in this area, we know an equilibrium solution exists for such N-person games, though actually finding an equilibrium is a highly non-trivial task (we recall Nash's proof of his theorem is a non-constructive theorem). There do exist a number of heuristic methods to estimating and finding the equilibrium solutions of N-person games, but *in this research, we propose a new method of solving N-person games inspired by the Brouwer's fixed point theorem (a crucial ingredient in Nash's doctoral thesis), and show how such methods can be employed to optimally manage assets of a sensor system in competition, coordination and cooperation against a common adversary.*

Finally, *we present a sensor network simulation environment with sensor resource management (SRM) algorithms that brings the three aforementioned ideas together in a hierarchical and dynamic way, which not only validates our ideas but also can serve as a predictive tool of what the adversary may do in future and what the corresponding defensive strategy should be.* Our approach is based on realization that there is a natural hierarchical breakdown between a 2-person game that models the interaction between a sensor network and its common adversary; and an N-person game that models the competition, cooperation, and coordination between asset of the sensor network. We have developed a concept of operations as well as SW prototypes to simulate such an environment to corroborate our theory.

The ultimate goal of developing these approaches is that they would be used for improving the importance of Space Situational Awareness (SSA), whose importance cannot be overemphasized. In fact, according to 2005 Quadrennial Defense Review (QDR), the United States must have "unfettered, reliable, and secure" access to its space assets, assured, for now, by "improving space situational awareness and protection, and through other space control measures." [1] This echoed the findings of the earlier 2001 version, which strongly insisted that "the US must not only exploit the advantages of the "high ground" of space, but that it also should develop a robust means to deny the use of space assets to any adversary." We strongly believe one rigorous theoretical foundation to improve SSA is by means of game theory, and it is our ardent hope that the new approaches that were developed through this research will make a positive contribution in this regard.

2. THEORY

2.1. CONOPS, GAMES, OVERALL APPRAOCH

Advanced optimization-based algorithms for Sensor Resource Management (SRM) have been a research focus area for multi-sensor ISR systems. However, these algorithms usually offer the potential for automating the sensor control process in response to level 1 sensor data fusion (object or track-level) estimates, even though studies have indicated that such SRM algorithms may have limited value because the algorithms are optimized for track maintenance without any assessment of overall situation context. To overcome such inadequacies, we had begun developing a framework for representing the expected information value of planned sensor measurements as it contributes to higher-level situational inferences, in our previous SBIR Phase II effort called RESE. In this effort, we developed a hierarchical valuation model that estimates target value on the basis of Level 2 fusion information (group identity) as well as Level 1 information, and showed that the optimization algorithms that maximizes this newly augmented value function gave better results not only for classifying group identity but also for track qualities of targets on the ground.

One of the most important purposes for multi-sensor ISR system, however, is to infer the adversarial intent given the current estimate of the adversarial observables, which is the focus of level-3 information fusion. This is a hard problem, especially given that such intent changes over time and depends on the enemy's perception of the blue force's intent. However, as we briefly saw at the end of our RESE effort, it is possible to use one mathematical framework, namely game theory, to make such reasoning rigorous and effect a high-level data fusion, which utilizes all level-1, level-2 and level-3 information.

Buoyed by this encouraging yet ad-hoc limited result near the end of RESE effort, we decided to find a theoretical foundation upon which we would be able to explain these results, which led us to the discovery of Kalman-filtering techniques for 2-person dynamic game through this current research. Furthermore, aware of the current and future operational need to manage heterogeneous set of multiple sensors in a highly dynamic, distributed, and decentralized environment, we again looked to game theory (this time N-person game theory), treating the sensors in the network, each endowed with different capabilities, modalities, and communication bandwidth, as a player in a N-person game (N-person coalitional game, especially if N becomes large). This has the advantage of providing common framework where the needs and capabilities of different sensors can be compared and coordinated, allowing multiple sensors to work together through binding agreement to achieve the most optimal cooperative resource allocation.

Figure 2 shows how our overall approach will work. To begin with, as two-person game is defined by a set of strategies for each player and the payoff matrix which assigns a value that each player perceives for himself given an ensemble of strategies that each player has made, we will begin by defining an overall adversarial space interaction between blue and red in order to turn this into a 2-person game, where blue is our sensor network and red is the network of adversarial entities. Though our setup is general enough for either zero-sum game or bi-matrix game, we have chosen a zero sum game for the sake of simplicity. Then, by solving such a game for the adversarial strategy as well as the blue strategy, we attempted to infer as well adversarial intent as well as how the sensor network should respond to such an intent.

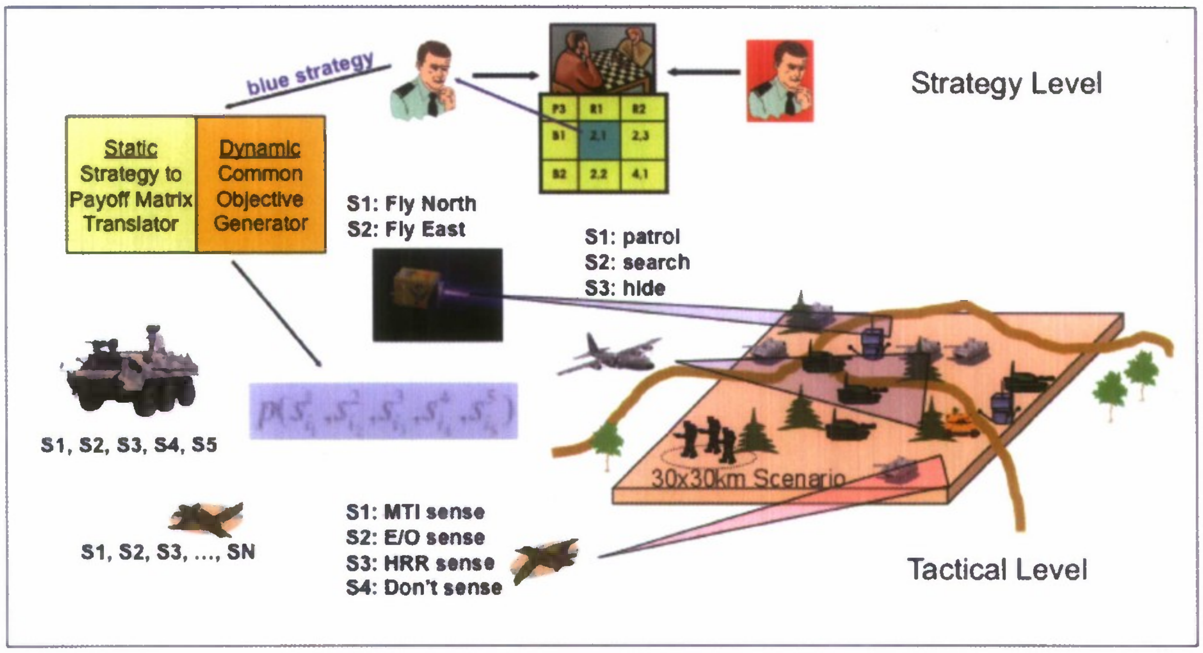


Figure 2: Hierarchical Dynamic-Game Theoretic Approach

Inevitably, such definition of 2-person games will depend on many factors, which come with unavoidable variability (environmental as well as human), and thus, we will use the theory of uncertain game (otherwise known as theory of games with incomplete information) throughout our effort. This means that the games we consider actually become a probability distribution over the space of possible games, allowing us to experiment over a number of different games that model the interaction between the offense and defense, one by one or even all at once.

We will then use the innovative approaches mentioned in the previous and expounded further in the following section to build our hierarchical dynamic game-theoretic SRM approach, which is depicted schematically in Figure 3. As we described above, the 2-person game that models the interaction between the offensive entity and defensive sensor network, which results in the common strategy that the sensor assets need to achieve together, in response to the strategy that the adversary will likely adopt. Such a common strategy is then passed onto different sensor assets at which point Common Asset Coordinator (CAC) will then solve an N-person game in a competitive, coordinated, or a cooperative fashion for an equilibrium solution which will give rise to a strategy that each asset will need to adopt, in order to ensure optimal resource allocation among all the assets. Finally, each sensor asset, coupled with its GMTI tracker, forms a TSP (Tracker Sensor Pair) and locally employs a 2-person dynamic game of its own in order to make its most optimal local decision, which then makes a decision about whether to choose this local decision or the decision produced from the N-person game or a combination of both. We now describe this in more detail in the next section, and we begin by reviewing the result of the RESE effort.

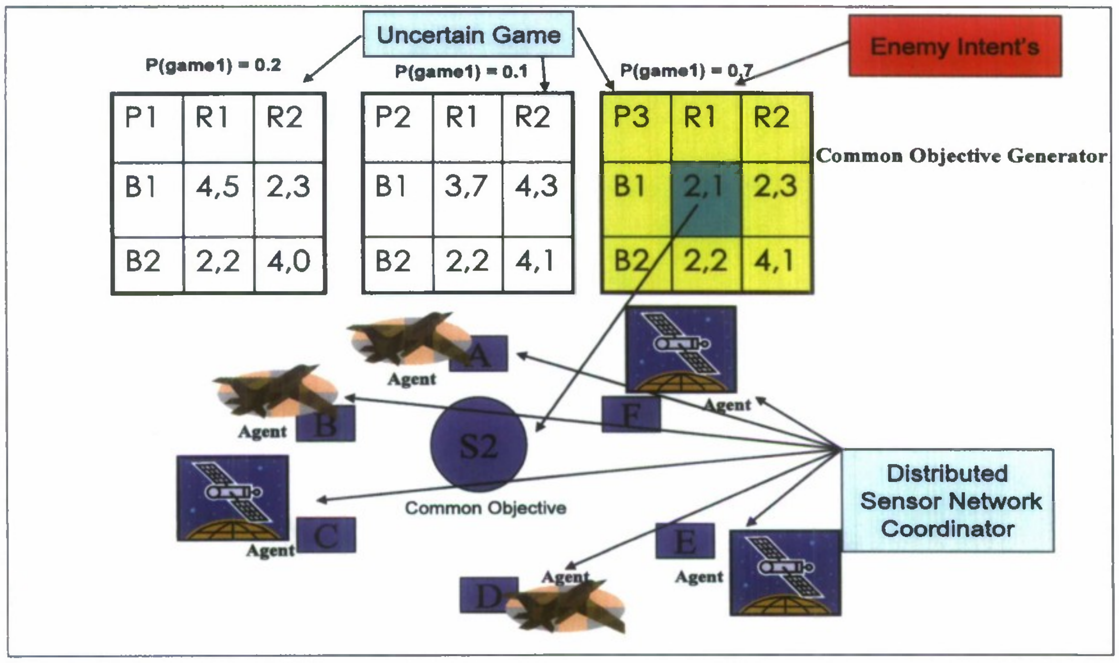


Figure 3: Interplay between 2-person game and N-person Game

2.2. DEVELOPMENT OF EFFICIENT TWO-PERSON GAME SOLVER

We have developed a prototype software tool, GameSolver, for solving mathematical games that are relevant for this research effort. Of course, the most crucial aspect of using game theory is the determination of equilibrium solutions (i.e. Nash Equilibrium solution). This is usually a difficult problem, but for a two- person, zero sum game, i.e., blue's gain equals red's losses and vice versa, there are known solution methods for specific payoff matrix structures (combinatorial solution methods exist also for 2-person non-zero sum games, but we concentrated our efforts to zero-sum games for the sake of simplicity). More specifically, these solution methods are adaptable to the form of the payoff matrix:

- *2 x 2 matrix* - This is the simplest case. For example, the enemy has two strategies: wait or attack, and the blue force has two strategies: wait or defend. For a payoff matrix of $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, the solution can be written explicitly:

$$p_1 = \frac{d-b}{a+d-b-c},\ p_2 = 1 - p_1$$

$$q_1 = \frac{d-c}{a+d-b-c},\ q_2 = 1 - q_1$$

where (p_1, p_2) represents the blue force mixed strategy solution, i.e., the blue force should adopt strategy 1 and strategy 2 with probability p1 and p_2, respectively; similarly for the red force and (q_1, q_2).

- *Severely Asymmetric matrix (m x 2 or 2 x n)* – This describes when one side has a much richer set of strategies than the other. This often describes situations in unconventional, asymmetric warfare in this age of GWOT, where insurgents and terrorist are known to have a wide variety of strategies against more conventional blue force strategies. These types of games can be readily solved with graphical analysis on the set of linear functions of two variables (i.e. lines).
- *Square payoff matrix (m x m)* - The offense and defense have an equal number of strategies. However, unlike the 2 x 2 case, there is no explicit solution formula for blue and red strategies. Instead, the well-known *principle of indifference* can be used to set up a linear system of m equations with m unknowns.
- *Non-square payoff matrix (m x n)* – The most general case where one's number of strategies is not limited by one's opponents. In this case, we follow the clever trick first realized by G. Owen in 1982 to turn this problem into a linear programming problem, and then apply the well-known simplex method.
- *Invariant games* - The payoff between blue and red do not change under some 1-1 mapping of blue and red strategies onto themselves. In this case, the Nash solution is also invariant, thus making the solution easier to find. We should first reduce the game by the mapping.

Figure 4 summarizes our approach:

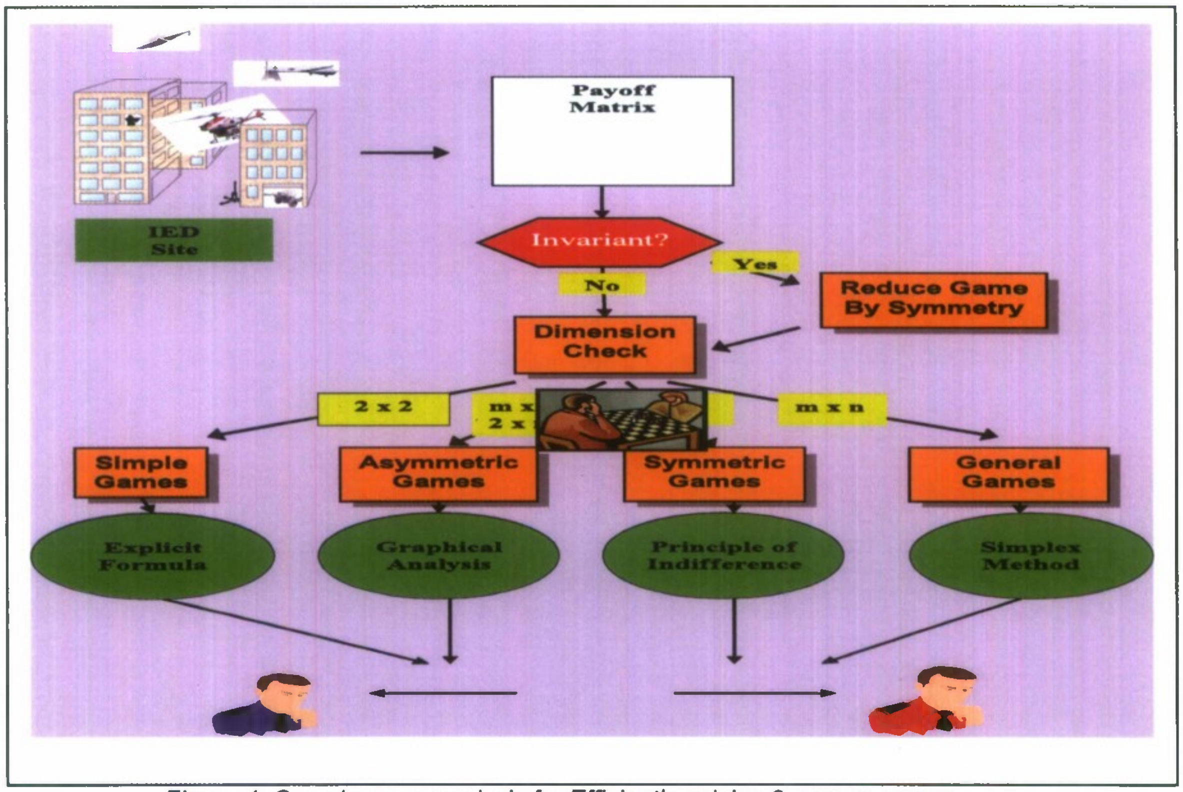


Figure 4: Case by case analysis for Efficiently solving 2-person games

2.3. FILTERING TECHNIQUES FOR SOLVING DYNAMIC GAMES

As much as the two-person game theory aptly describes the adversarial situation that must be considered, in particular, for space situation awareness, and as important as Neumann's and Nash's equilibrium results of two-person games are, they are fundamentally about static games. Therefore, they do not easily give insights to a dynamic situation where the game is repeated many or even infinite number of times (repeated games), where the memory of players add dynamics to the games, or where the rules of the games themselves change (fully dynamic games).

There are theorems, such as Folk Theorem [2], which proves that an infinitely repeated game should permit the players to design equilibriums, supported by threats, with outcomes being Pareto efficient. Also, at the limit case of continuous time case, instead of discrete time, we may bring to bear some techniques of differential games, which are inspired mostly by well-known techniques of partial differential equations applied to Hamilton-Jacobi-Bellman equation, as pioneered by Rufus Issacs [3]. However, unfortunately, these approaches do not readily give insights to how to select and adapt strategies as the game changes from one time epoch to the next, as is necessary in order to gain space situational awareness in all of its dynamics.

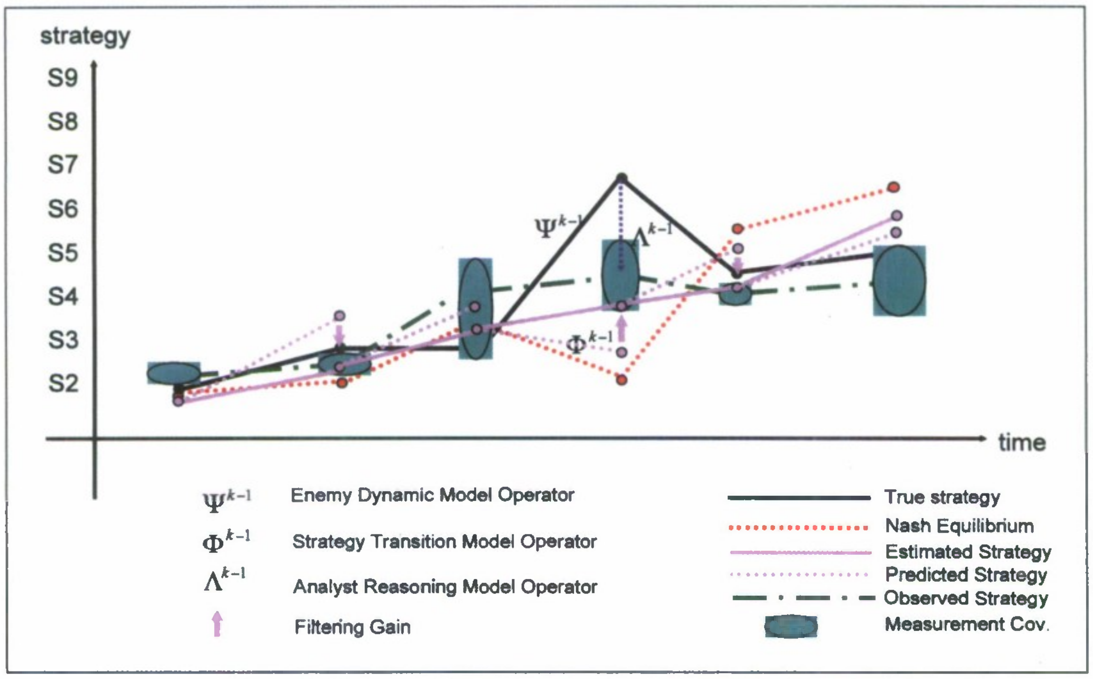


Figure 5: Pictorial View of Filtering Techniques for Dynamic Games

Therefore, based on our experience in Kalman filtering techniques for estimating states of moving targets, we propose *a different approach to solve dynamic games in order to help figure out the intent of an adversary to gain superior space situational awareness*. The key observation in our approach is, *ironically speaking*, not to be overly concerned about Nash equilibrium. We recall that Nash equilibrium is useful when a player does not know what the other player may do, and thus it is some sense a safe strategy that each player in a game can adopt. However, when a sensor network is employed to gain awareness of a situation in space, each time a measurement is made by a sensor (kinematic state, classification state, etc., depending on the sensor modality), a measure of understanding about the adversary is gained, however incremental it may be. Our approach exploits this incremental gain each time a sensor in a sensor network is employed in order to solve the dynamic game in question and gain the best estimate of the intent the adversarial space entities, which may still continue to change.

Our approach is inspired by our experience with Kalman filtering, as described pictorially in Figure 5, where *y*-axis represents adversarial strategy and *x*-axis represents time. As in Kalman filtering paradigm, our approach tries to find the right balance in combining the prediction of what the adversarial intent may next be, governed by a model of adversarial strategy transition (operator Φ^{k-1} in the figure) and the most recent measurement of the adversarial intent, given by another model of how an analyst may process the sensor data to map the sensor measurement to a given set of adversarial strategies (operator Λ^{k-1} in the figure). However, unlike Kalman filtering, a third factor is also balanced with the prediction of adversarial strategy and the measurement of adversarial strategy, namely Nash equilibrium strategy for the adversary (the strategy that the adversary will most likely adopt without any further insight into friendly force's intent).

Mathematically, this insight translates into adding a third-term to the famous Kalman filtering equations. Let us start from the original Kalman filtering equations as below:

$$\hat{y}_t^t = \hat{y}_t^{t-1} + K_t(x_t - B\hat{y}_t^{t-1})$$
$$P_t^t = (I - K_t B)P_t^{t-1}$$
$$K_t = P_t^{t-1}B^T(R + BP_t^{t-1}B^T)^{-1}$$

where

$\hat{y}_t^t, \hat{y}_t^{t-1}$ is state estimate of a target at time instance *t*, given observation up to time *t*, *t-1*

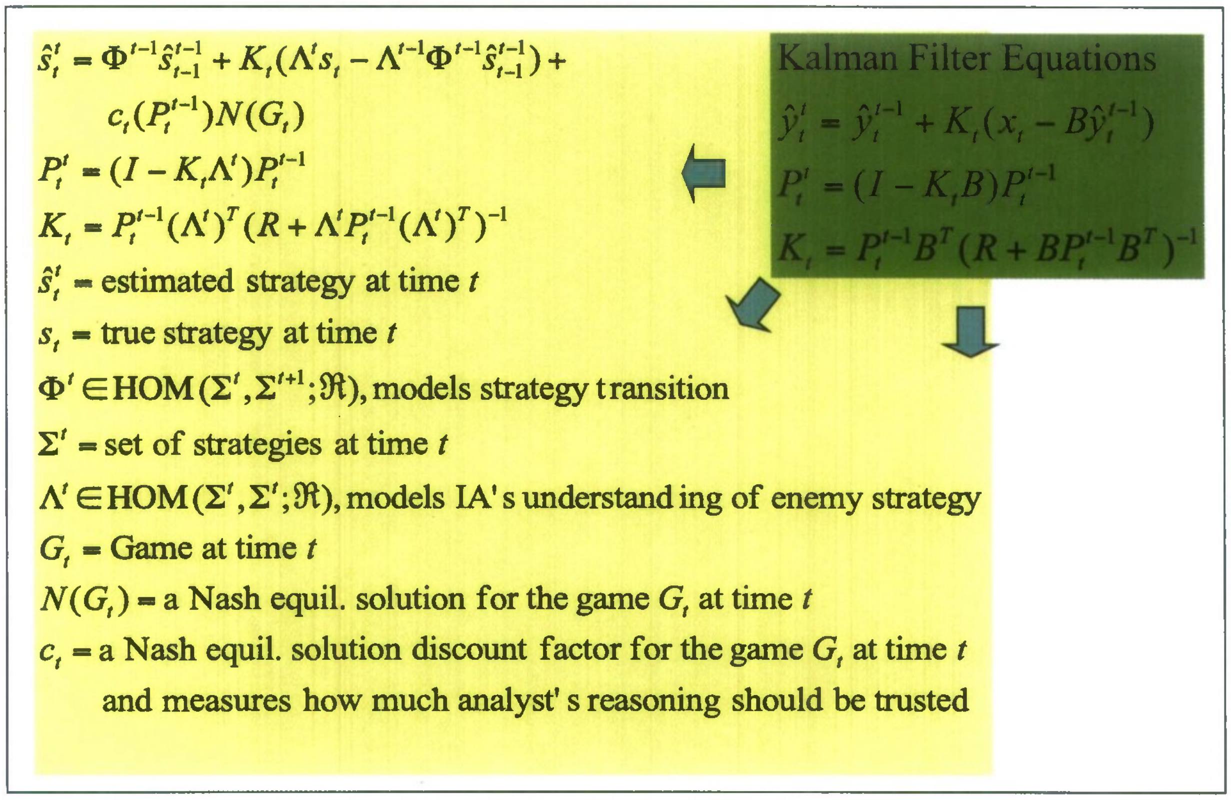
x_t is the observation at time instance *t*

$\hat{P}_t^t, \hat{P}_t^{t-1}$ is the covariance at time instance *t*, given the measurement up to time *t*, *t-1*

K_t is the Kalman gain at time instance *t*

B is the measurement operator

Starting from these equations, we derive the following set of equations, described in the picture form below in Figure 6:



$$\hat{s}_t^t = \Phi^{t-1}\hat{s}_{t-1}^{t-1} + K_t(\Lambda^t s_t - \Lambda^{t-1}\Phi^{t-1}\hat{s}_{t-1}^{t-1}) + c_t(P_t^{t-1})N(G_t)$$

$$P_t^t = (I - K_t\Lambda^t)P_t^{t-1}$$

$$K_t = P_t^{t-1}(\Lambda^t)^T(R + \Lambda^t P_t^{t-1}(\Lambda^t)^T)^{-1}$$

$\hat{s}_t^t$ = estimated strategy at time t

s_t = true strategy at time t

$\Phi^t \in \mathrm{HOM}(\Sigma^t, \Sigma^{t+1}; \Re)$, models strategy transition

Σ^t = set of strategies at time t

$\Lambda^t \in \mathrm{HOM}(\Sigma^t, \Sigma^t; \Re)$, models IA's understanding of enemy strategy

G_t = Game at time t

$N(G_t)$ = a Nash equil. solution for the game G_t at time t

c_t = a Nash equil. solution discount factor for the game G_t at time t and measures how much analyst' s reasoning should be trusted

Kalman Filter Equations

$$\hat{y}_t^t = \hat{y}_t^{t-1} + K_t(x_t - B\hat{y}_t^{t-1})$$

$$P_t^t = (I - K_t B)P_t^{t-1}$$

$$K_t = P_t^{t-1}B^T(R + BP_t^{t-1}B^T)^{-1}$$

Figure 6: Mathematical View of Filtering Techniques for Dynamic Games

The first question in Figure 6:

$$\hat{y}_t^t = \Phi^{t-1}\hat{s}_t^{t-1} + K_t(\Lambda^t s_t - \Lambda^{t-1}\Phi^{t-1}\hat{s}_{t-1}^{t-1}) + c_t(P_t^{t-1})N(G_t)$$

shows how $\hat{S}_t^t$, next estimate of the adversarial intent given by its next strategy, is a combination of three terms, namely:

- $\Phi^{t-1}\hat{s}_t^{t-1}$: prediction of the next adversarial strategy given strategy transition model Φ^t
- $K_t(\Lambda^t s_t - \Lambda^{t-1}\Phi^{t-1}\hat{s}_{t-1}^{t-1})$: measurement of the current adversarial strategy given IA (Intelligence Analyst)'s model of process sensor measurement data
- $c_t(P_t^{t-1})N(G_t)$: Nash equilibrium tempered by a discount factor

Furthermore, the next two equations describe how uncertainty of adversarial strategy (given by the covariance term P_t^t) and Kalman gain (K_t) grow under this dynamic system. To gain further understanding of this dynamic system, we plan to ask the following questions (& more as we continue the research) :

1) Does $\hat{S}_t^t$ converge to true strategy?
2) If the answer to 1) is yes, how fast does it converge?
3) How stable is this dynamic system?
4) What is a natural non-linear extension of this system? (e.g. what is its analogue to scented Kalman filter)?

Though convergence and stability of these new dynamic equations above still need to be proved, through a prototypical simulation environment that we have used in this research, we have a preliminary empirical verification the effectiveness of these techniques though an extensive Monte-Carlos runs, as we will see in later sections. We plan to focus our future research on these questions we raised above, and in so doing, we solidify the theory of our filtering techniques for dynamic games and present a practical and computationally feasible way to understand adversarial interactions for future applications.

2.4. BROWDER'S FIXED POINT APPORACH FOR N-PERSON GAMES

Through this research effort, we also developed innovative approaches to solve N-person game, for which there is no known general approach, as Nash's famous result on equilibrium was an existence proof only, not a constructive proof. The key idea for N-person game is not to resort to some heuristic approach in order to solve N-person games, as it is usually done for static N-person game, but rather to use the techniques inspired by Browder's Fixed Point theorem (Figure 7) used in Nash's original Ph.D. thesis. For example, Nash's arguments rest upon the following equations [5] that describe a transformation from one strategy $\zeta = (s_1, s_2, s_3, \ldots, s_n)$ to another $\zeta' = (s'_1, s'_2, s'_3, \ldots, s'_n)$ by the following mapping:

Pictorial view of Browder's fixed point theorem

Thm (Browder): *A continuous function from a ball (of any dimension) to it self must leave at least one point fixed.*

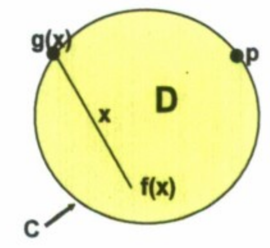


Figure 7: Fixed Point Theorem for B^2

$$s'_i = \frac{s_i + \sum_{\alpha} \varphi_{i\alpha}(\zeta)\pi_{i\alpha}}{1 + \sum_{\alpha} \varphi_{i\alpha}(\zeta)}$$

where

$\varphi_{i\alpha} = \max(0, p_{i\alpha}(\zeta) - p_i(\zeta))$

ζ is an n-tuple of mixed strategies

$p_i(\zeta)$ is the corresponding strategy to player *i*

$p_{i\alpha}(\zeta)$ is the payoff to player *i* if he changes to α^{th} pure strategy

We note the crux of Nash's proof of the existence of equilibrium solution(s) boils down to this fixed theorem (see a pictorial explanation description of the theorem above) applied to the following mapping $T : \zeta \rightarrow \zeta'$ and by investigating the delicate arguments that Nash used to convert these fixed points into his famous Nash equilibrium solutions, we also plan to use these arguments to find a constructive methods to find the solutions of N-person games. One possible approach we have in mind is to first start from a set of sample points in the space of strategies, and compute:

$$C(\zeta) = \frac{\|T(\zeta)\|}{\|\zeta\|}$$

$C(\zeta)$ measures the degree to which a strategy ζ is changed by the map $T : \zeta \to \zeta'$. We will look for a strategy ζ where $C(\zeta)$ is close to 1, as the possible initial search points to look for equilibrium points to start such iterative search process.

2.5. FICTIOUS PLAY FOR DYNAMIC N-PERSON GAMES

Whenever sensor nodes can establish effective communication with each other, we propose to use the theory of bargaining, well understood in game theory community, to model their cooperation and coordination among sensors in order to achieve the most optimal solution (Praeto optimal), which may not be sometimes achievable in non-cooperative games.

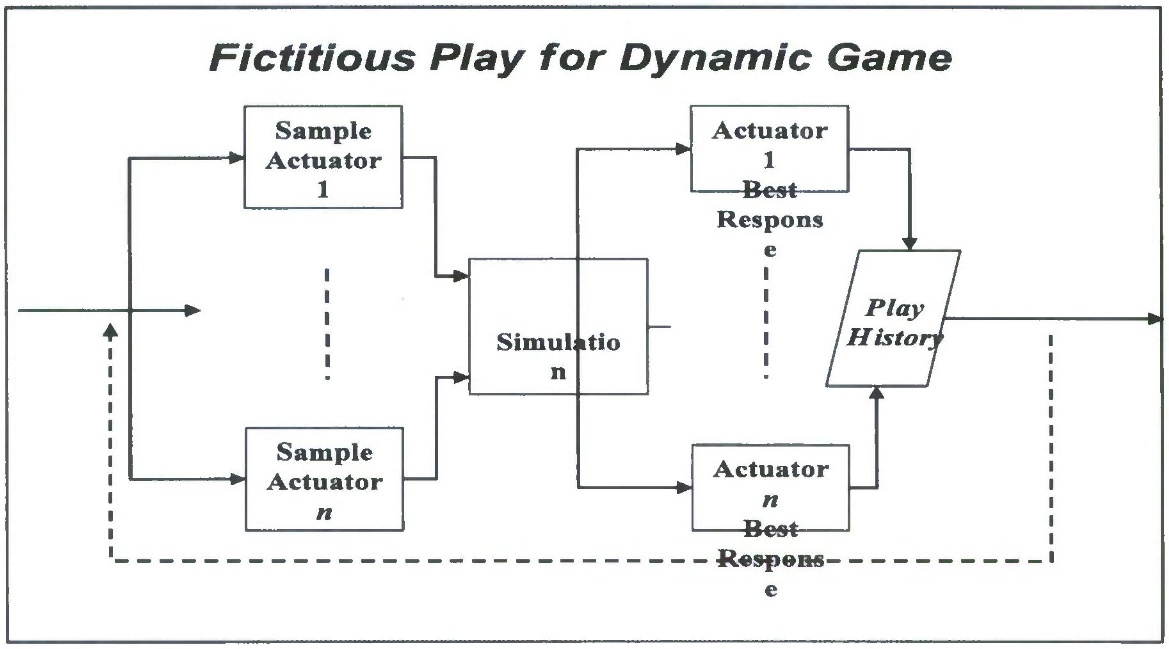


Figure 8: Fictitious Play for N-person games of Patek, .et al.

Furthermore, for dynamic N-person games with a common objective among the players, we plan to investigate for future research the applicability of the techniques that Prof. Steve Patek and his colleagues at the University of Virginia have developed for generating endogenous Nash equilibrium, which is schematically depicted in Figure 8, using fictitious play learning algorithm.

In considering optimal architecture for command and control of large system (which is a key issue for any future sensor systems), Dr. Patek and his colleagues have studied the use of game theoretic techniques in distributing the search for global extrema in large-scale, discrete stochastic optimization problems. In their work, they attribute "player" status to each asset involved in the search for a globally optimal solution, each having a utility function equal to the objective function of the optimization problem being solved. Then, generalizing the "fictitious play" algorithm of game theory, they engage all players in a repeated process of "best responding" to the past search activities of all other players along with their observations about what constitutes an optimal solution. In applying this algorithm to resource allocation problems in a variety of application domains, it was observed that the algorithm converges in the sense that all players will eventually converge on a pure strategy Nash equilibrium, corresponding to a globally optimal solution when the underlying objective function is convex. Under suitable regularity assumptions, this result holds in general, and we plan to explore further the suitability of fictitious play-based optimization algorithms for managing sensor assets. Through such investigation, we hope to understand coordination needed among sensors to respond to threats exogenous to the network, especially for the purpose of distributed resource management for sensors.

2.6. SRM ENVIRONMENT FOR HIGH LEVEL FUSION (LOCAL)

Figure 9 shows how we may put together some of the approaches we described so far to devise a local version of hierarchical dynamic game-theoretic sensor simulation environment, which we started in our RESE effort. We now describe this in more details, making connection with multi-level date fusion:

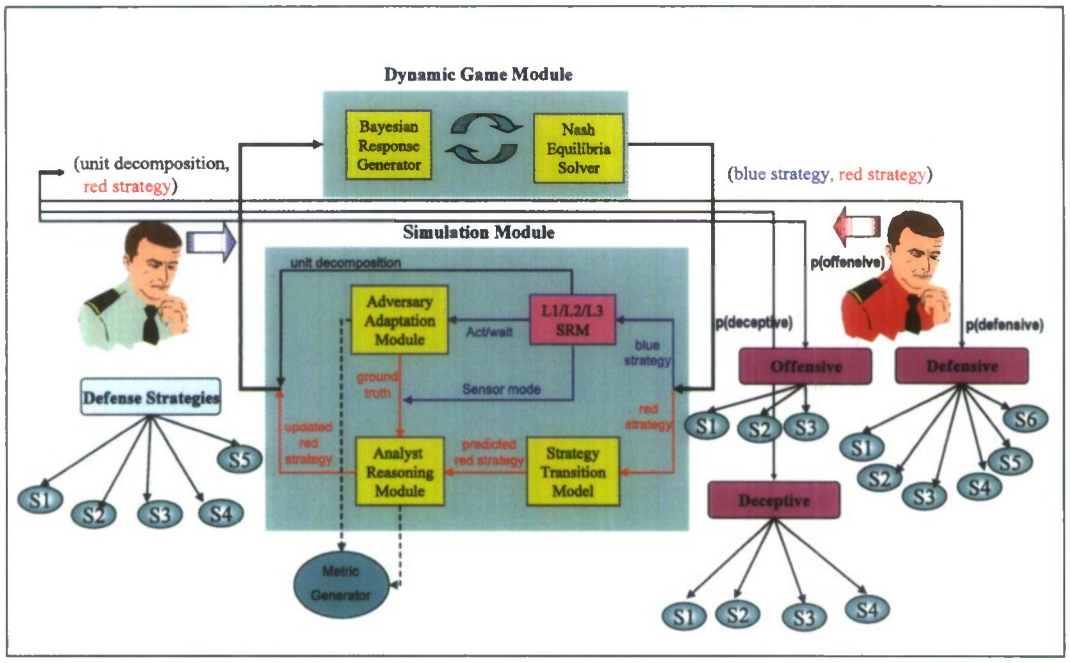


Figure 9: Sensor Simulation Environment

2.6.1. From Level 1 to Level 2 via Statistical Modeling

The key connection between level 1 and level 2 fusions is an estimation technique that we have developed to determine the identity of a *group* of objects, given the identities of the individual objects themselves. The basic paradigm is as follows:

- We are given a set of *measurements,* associated in a non-redundant manner with *objects.*
- Objects are known to submit to a finite number of distinct and exhaustive *classifications.*
- Objects are known to co-occur in *groups*, where each group is defined uniquely by its object composition.

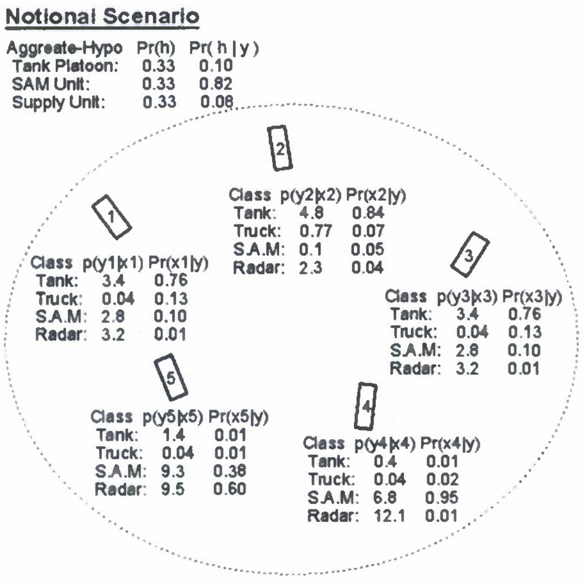


Figure 10: Notional scenario for group and target classification

Given prior knowledge on the relative frequency or probability of specific object groupings, how are we then to judge which hypothesis of group identity best explains the available object-level observations? We must also consider the impact of this prior knowledge upon the classification of individual objects. That is, how should the context in which an object is observed (the presence of other observed objects) affect our classification of that object? More precisely, given the following:

- $o1, \ldots on$: the set of n objects in the group G
- $x1, .., xq$: possible object types
- $y = (y1, .., \underline{y}n)$: the set of object measurements on the n objects
- h: a group type hypothesis on group G,

and assuming that we have per-object measurement likelihoods conditioned on the targct typc p(yi | xj), we would like to (a) evaluate the probability of various group hypotheses Pr(h | y), and (b) evaluate the updated target probabilities Pr(xj | y) (see Figure 10 for a notional scenario). Wc tackle the problem by laying out a precise probabilistic model for the occurrence and observation of groups of objects. This model consists of three components, namely (i) a prior model, (ii) a detection model, and (iii) a measurement model. More precisely:

(i) The Prior Model: The prior model specifies what groups of objects may occur and the relative frequencies with which these groups occur. Group hypotheses are spccified by the group composition, i.e., the number of instances of each object class are present within the group.

(ii) The Detection Model: Our observation of a group may bc incomplete in that thc objccts themselves may be difficult to identify. To model our uncertainty as to whether or not the "correct" objects were detected we introduce a detection model, using the Poisson process based upon the probability of detection, false alarm rate that is consistent with expected clutter density.

(iii) The Measurement Model: The measurement model specifies the probabilistic distribution of measurements conditioned upon each of the possible target classifications. Ambiguity as to the correct classification of an observed object is created by overlap of measurement modcls for different object classes.

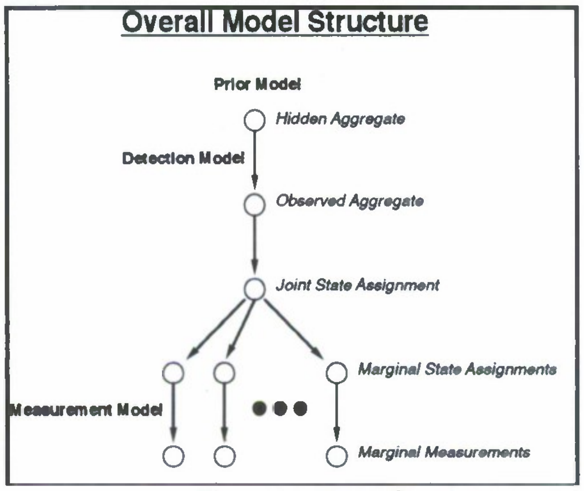


Figure 11: Model Structure

These three processes may be thought of as being "piped" in that the output of each fecds into the next, as illustrated in Figure 11. The trce diagram shown there illustrates the top-down structure of our model. The prior model specifies the prior probabilitics of the states at the root node. The Detection Model determines thc transition probabilitics from the hidden aggregate to the detected aggregate - fundamental considerations determine the probabilistic mapping from the observed aggregate state to the various object classifications. Finally, the measurement model gives the transition probabilities from an object-classification node to the associated measurement.

The calculation of likelihoods under this model is then summarized as follows. Using Bayes' rule, the probability of an aggregate hypothesis (h) conditioned upon all observations (y) is

$$\Pr(h \mid y) = \frac{p(y \mid h)\Pr(h)}{p(y)}$$

where Pr(h) is given by the prior model, $p(y \mid h)$ is the likelihood of the measurements conditioned upon h, and $p(y)$ is a normalizer:

$$p(y) = \sum_h p(y \mid h)\Pr(h)$$

Using the detection model, the mapping from the aggregate hypothesis, h, to the detected distribution, d, may be computed by calculating transition probabilities, $\Pr(d \mid h)$. Using these transition probabilities and the chain-rule we have:

$$p(y \mid h) = \sum_{d \in D(n,q)} p(y \mid d)\Pr(d \mid h)$$

where the summation is taken over $D(n,q)$, the set of all possible distributions of n objects into q classes, where n is the number of detected objects.

Finally, using the measurement model, we compute the likelihood of a set of observations y conditioned upon the distribution of detected objects, d, by direct calculation using the chain-rule as:

$$p(y \mid d) = \sum_{a \in A(n,q)} p(y \mid a)\Pr(a \mid d) = \frac{1}{|\Omega(d)|}\sum_{a \in \Omega(d)} \prod_{k=1}^{q} p(y_k \mid a_k)$$

The former sum is over the set of all possible assignments of q object-classifications to n objects. The latter formula indicates the "brute-force" method of calculating the measurement likelihoods conditioned upon a distribution hypothesis. The summation is over all assignments a consistent with the hypothesized distribution d. In general, this computation could be very time consuming and may not be feasible. To avoid that, we propose to develop an approximate evaluation process by collapsing the entire object classification probability distributions into one aggregated "group classification vector" and skip the enumeration of an exponentially growing set in the joint state assignment.

Impact of Group Hypotheses on Value Functions for Sensor Resource Management

Given the set of track-level measurements y and the probability distribution over all group hypothesis $\{\Pr(h \mid y)\}h$, we can construct a value function ($VL12$) for the group that incorporates both level 1 and level 2 fusion measurements:

$$V_{L12}(G \mid y) = \sum_h V(h)\Pr(h \mid y) \sum_{j=1,..n} V(o_j \mid y_j)\Pr(xj \mid y)$$

In this value function, the first summation represents the level 2-fusion value, where *V(h)* is the commander's preference (importance) of a particular group type. The second summation represents the level 1 fusion value, where $V(oj \mid yj)$ is the value of object *j* given *yj*, the object level measurement of object *j* (note that this valuation is itself determined via the commander's preference on the object type as well as the target probabilities $\Pr(xj \mid y)$).

To utilize this value function for sensor resource management, we rely on sensor models for generating *expected* object level measurements *y*, thereby yielding varying level 1 and level 2 values, hence varying total value, for alternative sensors and sensor modes. This value function can thus be used to determine sensor resource allocation decisions that reflect the commander's preference in terms of object or group types as well as the capabilities of the various sensors available.

2.6.2. From Level 2 to Level 3 via Uncertain Games

Exploitation of level 1 and level 2 (red force) data tends to be a local (in time and space) process that analyzes data within a particular field of view for a short period of time. In contrast, level 3 data exploitation, determining the intent of the enemy, tends to be a more global process (longer time-horizon, looking at the entire AOI). Inferring the red force intent is a difficult problem, especially given that it changes over time and depends on the enemy's perception of the blue force intent. As we argued thus far, we have used one mathematical framework for making such reasoning rigorous, namely game theory.

However, we found that just a textbook static 2-person game in the spirit of von Neumann wasn't quite sufficient for our research. This was due to the uncertainty in our problem domain which stems from the uncertainty in the red force identity, which then has a direct bearing on the red force intent. Different units do have different strategies and payoff functions, which consequently define different games with different equilibrium solutions. To illustrate, consider the example shown in Figure 12, where the blue force must choose among five possible strategies against a red force whose identity is unknown but is assumed to be either a defensive, offensive, or supply unit. The red force unit would have different strategy choices depending on its identity - for example, six possible attack strategies for an offensive unit, three counter measures for a defensive unit, and three transport strategies for a supply unit. Given the uncertainty of the red force identity and the assumed possible strategies for each red force unit type, the blue force must choose its strategy that will be most successful.

To resolve the red force identity uncertainty and thus the identity of the game, we have used the output of the level 2 data exploitation, in particular $\{\Pr(h \mid y)\}_h$, the probability distribution over all group hypothesis *h* given the track-level measurements *y*. This is how our level 2 analysis and algorithms are related to those of level 3.

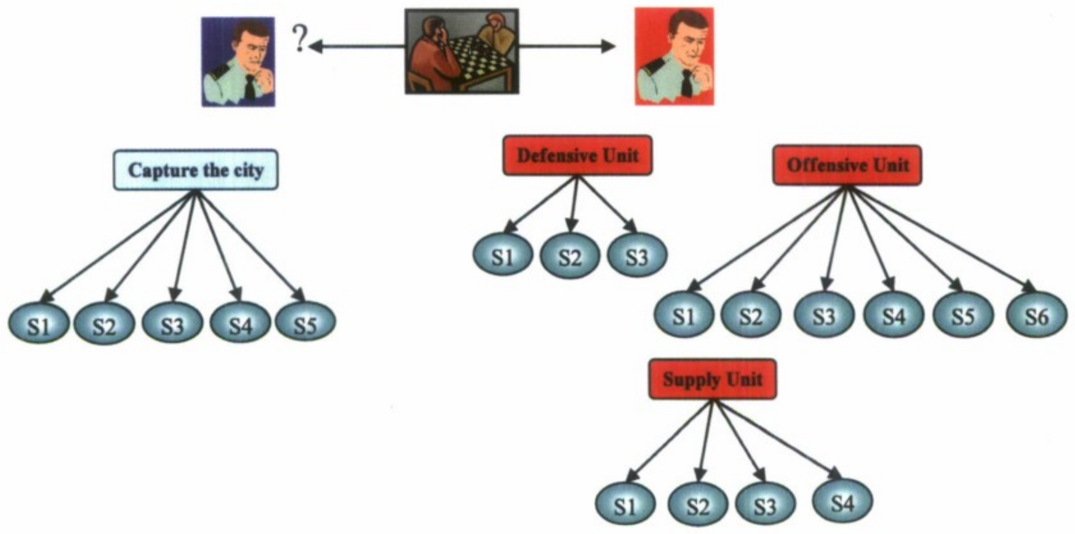


Figure 12: The blue force must determine which strategy to adopt unsure of the identity of the red force

Under an assumption of the red force identity, we can define the two person game that the blue force and red force should play, for which we know that a Nash solution exists, thanks to the work of von Neumann and Nash. Therefore, *we first develop a mapping between red force unit types and the set of strategies and related payoff functions, which define the game.* As contextual information such as weather, doctrines, and terrain information will affect red force strategies and payoffs, such information will be taken into account in this mapping, in future research.

Next, given the game defined by this mapping, it still remains to determine the Nash equilibrium of the game, and thus the intent of the red force. This is in general a difficult problem. However, in the special case of two-person non-zero sum game as we have in this context, Mangasarian and Stone showed that the Nash equilibrium can be systematically found using techniques of quadratic programming. Further, given the uncertainty in the enemy's identity, our analysis will result in a Nash equilibrium solution, i.e., an intent distribution, for each of m defined games, each game corresponding to an assumption (hypothesis) on the red force identity (m=3 in the example in Figure 12). Though this set of intent distributions can then be incorporated back into our hierarchical value function, we have set aside this for future research.

2.6.3. SW Prototype Architecture for Game theoretic L1/L2/L3 Fusion

Figure 13 shows a prototypical view of the software architecture for the simulation environment that we have built which simulates game theoretic multi-level fusion. L1/L2/L3 SRM module is as we described already in 2.6.1 and 2.6.2. We now proceed to describe the rest of modules in more details.

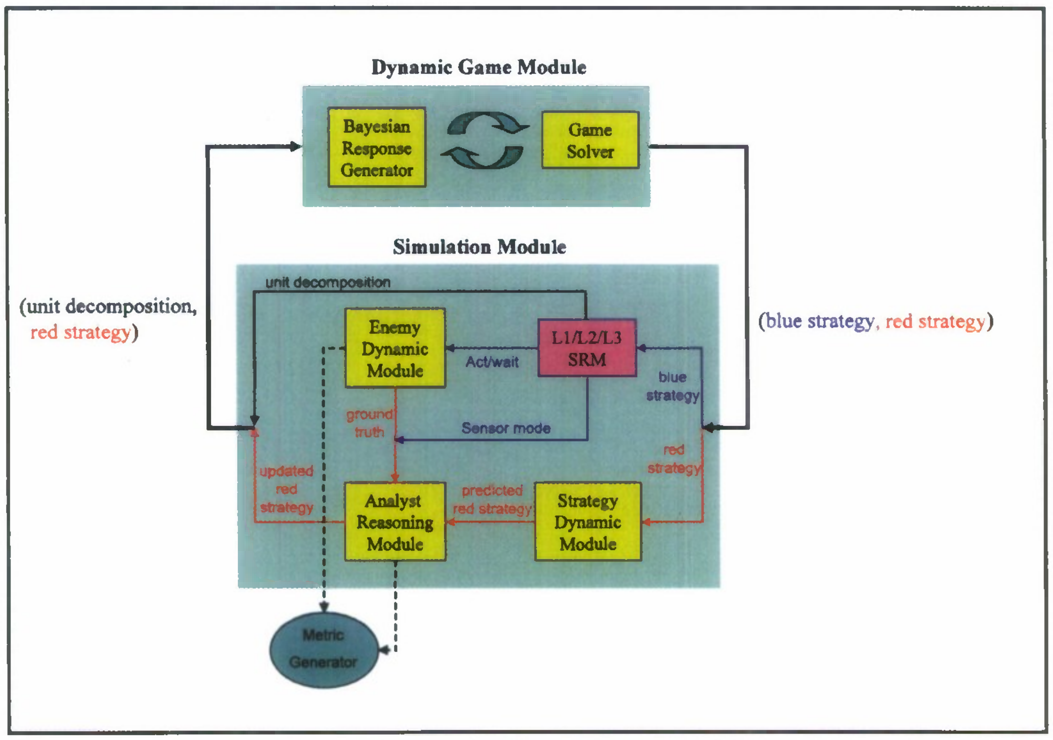


Figure 13: Prototypical SW Architecture for Game Theoretic SRM for High Level Data Fusion

Dynamic Game Module

A finite game is defined by a set of strategies for each player and the payoff matrix, representing numerically how valuable each player views a particular set of strategy choices. The situation modeled here between the blue and red forces can be described as a two-person zero sum game (in future work we will consider extensions to non zero sum games). As we mentioned in Section 2.6.2, since the blue force is uncertain of the enemy type, we define a game between the blue force and each of the possible red brigade type. To simplify the simulation and ensuing analysis, without loss of generality, we will assume that each enemy brigade type has the same set of strategies. Namely, we have (for this simulation):

- Blue Strategies = {act, wait}
- Red Brigade Types = (offensive brigade, defensive brigade, deceptive brigade) where
  - Offensive Brigade: consists of 2 units of type A.
  - Defensive Brigade: consists of 2 units of type B.
  - Deceptive Brigade: consists of 1 unit of type A and 1 unit of type B.
- Red strategies = {attack, defend, deceive}
- Payoff Matrix:
  -

- $A(offensive) = \begin{bmatrix} 5 & -2 & 1 \\ -5 & 3 & 2 \end{bmatrix}$
- $A(defensive) = \begin{bmatrix} -1 & 2 & 2 \\ 1 & -1 & 1 \end{bmatrix}$
- $A(deceptive) = \begin{bmatrix} 2 & -1 & 3 \\ -2 & 1 & -2 \end{bmatrix}$

where the first and second row of each payoff matrix corresponds to the blue force ***act*** and ***wait*** strategies, respectively, while the first, second, and third columns correspond to the red brigade ***attack***, ***defend***, and ***deceive*** strategies, respectively.

To illustrate further, we look at the *A(offensive)* payoff matrix in more details:

- If the enemy offensive unit is *attacking* (column 1), the blue force *gains* 5 (in some measure of utility) if it employs its sensors to act, i.e., *sense* the attack (and the offensive unit *loses* 5). However, the blue force *loses* 5 if it fails to act when the offensive unit is attacking, while the offensive enemy unit *gains* 5 as it is attacking and thereby achieving its goal without being sensed.
- If the enemy offensive unit is *defending* (column 2), the blue force *loses* 2 if it employs its sensors to act in that (a) the blue force loses sensor resources and (b) it makes it easier for the enemy brigade to determine that they are being sensed (which may prompt the enemy to change its strategy – the dynamics of the enemy strategies will be explained more later). By waiting, the blue force saves sensor resources and avoids being observed, gaining 3.
- If the enemy offensive unit is *deceiving* (column 3), while there is some utility (gain of 1) in employing blue force sensors, it is of greater utility (gain of 2) for the blue force to wait before committing sensor resources given that the enemy is trying to deceive.

The payoff matrices for the defensive and deceptive red force brigades are similarly defined. The values of the entries of all the payoff matrices are determined somewhat arbitrarily as a means for generating proof-of-concept results in our simulation exercises. A rigorous methodology for determining the matrix values would require a thorough understanding of the red and blue force interactions and is itself a topic of considerable research and should be investigated as a top research for future research. In our current framework, however, the payoff matrices are treated as input to the simulation, and, therefore, different matrices define different games, one example of which is described above.

Bayesian Response Generator

Given the set of payoff matrices as defined above and the blue force estimate of the enemy strategy, represented by a probability distribution q (which is an input from L2/L3 simulation module), where:

$$q = (q_1, q_2, q_3)$$

and

q_1 = probability that the red force is executing the *attack* strategy

q_2 = probability that the red force is executing the *defend* strategy

q_3 = probability that the red force is executing the *deceive* strategy,

one can then calculate the *Bayes Response*, i.e., the *best* strategy that blue force should take. The Bayes response is the *mixed strategy p* in the space of mixed strategies *X** that achieves the following maximum:

$$\max_{1 \le i \le m} \sum_{j=1}^{n} a_{ij} q_j = \max_{p \in X^*} p^T A q,$$

where a_{ij} represents the (i, j)th element of the payoff matrix (a mixed strategy is a probability distribution over the strategy space, e.g., a 2-dimensional probability vector over the two blue force strategies in our case; a *pure* strategy is the special case of a unitary mixed strategy). As we are uncertain as to the enemy type (offensive brigade, defensive brigade, deceptive brigade), we compute the Bayes response for each of the three payoff matrices corresponding to the enemy type given the enemy strategy *q* as follows:

- *Bayes_p(offensive) = argmax(p)* $_{p \in X^*} p^T A(offensive) q$
- *Bayes_p(defensive) = argmax(p)* $_{p \in X^*} p^T A(defensive) q$
- *Bayes_p(deceptive) = argmax(p)* $_{p \in X^*} p^T A(deceptive) q$

Then we define the best response (Bayes response), *Bayes_p*, as follows:

*Bayes_p = Pr(offensive)*Bayes_p(offensive) + Pr(defensive)*Bayes_p(defensive) + Pr(deceptive)*Bayes_p(deceptive)*

where

- *Pr(offensive)* = probability that red is an offensive brigade, i.e. both units are of type A
- *Pr(defensive)* = probability that red is a defensive brigade, i.e. both units are of type B
- *Pr(deceptive)* = probability that red is a deceptive brigade, i.e., 1 unit each of types A and B.

These probabilities over brigade types are computed via the individual unit probabilities of the units forming each brigade, the output of the level 1/level 2 SRM module.

Game Solver

While the Bayes response provides the best response that blue should take given the current estimate *q* of the red force strategies, the red force may change its strategy assuming that the blue will adopt a Bayes response. Of course, anticipating such a change from the red, the blue may change strategies again, thereby prompting a red strategy change, and so on. Therefore, each side must determine the best strategy to adopt without knowledge of the opposing side's strategy choice. Assuming that each side acts rationally, the Nash solution provides the strategies that each side is most likely to adopt. For each of the three games defined above, the Nash equilibrium is determined as follows:

- For
  - $A(\mathit{offensive}) = \begin{bmatrix} 5 & -2 & 1 \\ -5 & 3 & 2 \end{bmatrix}$,

    $p(\mathit{offensive}) = \begin{bmatrix} \frac{8}{15} & \frac{7}{15} \end{bmatrix}$, $q(\mathit{offensive}) = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & 0 \end{bmatrix}$

- For $A(\mathit{defensive}) = \begin{bmatrix} -1 & 2 & 2 \\ 1 & -1 & 1 \end{bmatrix}$,
  - $p(\mathit{defensive}) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$, $q(\mathit{defensive}) = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & 0 \end{bmatrix}$
- For $A(\mathit{deceptive}) = \begin{bmatrix} 2 & -1 & 3 \\ -2 & 1 & -2 \end{bmatrix}$,
  - $p(\mathit{deceptive}) = \begin{bmatrix} \frac{2}{5} & \frac{3}{5} \end{bmatrix}$, $q(\mathit{deceptive}) = \begin{bmatrix} \frac{3}{5} & \frac{2}{5} & 0 \end{bmatrix}$

We then define *Game_p* and *Game_q*, the weighted Nash equilibrium over the three payoff matrices corresponding to the enemy brigade types, analogously as the Bayes response:

- *Game_p = Pr(offensive)*p(offensive) + Pr(defensive)*p(defensive) + Pr(deceptive)*p(deceptive)*
- *Game_q =Pr(offensive)*q(offensive) + Pr(defensive)*q(defensive) + Pr(deceptive)*q(deceptive)*

We next combine the Bayes Response and the weighted Nash solution as follows:

- $Output\_p = t \times Bayes\_p + (1-t) \times Game\_p$
- $Output\_q = t \times q + (1-t) \times Game\_q$,

where the parameter t is the measure of how much one wants to emphasize the current estimate of the enemy's strategy as opposed to the Nash equilibrium solution (this is akin to the usual technique in linear filtering theory, where *(Bayes_p, Bayes_q)* is an observation and *(Game_p, Game_q)* is a prediction). *(Output_p, Output_q)*, a pair of mixed strategies, is then sent to L2/L3 Simulation Module which we now discuss.

L2/L3 Simulation Module

The Dynamic Game Module provides *Output_p*, the blue force next sensor decision (action or no action), and *Output_q*, a probabilistic assessment of the enemy strategy. The L2/L3 Simulation Module utilizes these output strategies as follows:

Using the Blue Strategy

The blue force strategy (act or wait) is determined via the generation of a random variable against the output probability vector *Output_p*. In the case of action, we utilize the level I and Level II valuation function to select a sensor mode as described in 2.6.1. The current track quality state used in the level II valuation function for SRM includes a 24 Joint Kinematic and Classification (JKC) state Markov model. These JKC states discretize the tracking and classification quality into a finite number of possibilitics so that one could relate the sensor management decision to the valuation function in a compact and computationally efficient manner. In the case of no action, we extended our simulation algorithms to evolve the JKC state under the assumption of no look.

Using the Red Strategy

The assessed enemy strategy *Output_q* is used to predict the enemy strategy for the next time step via a two-step process. First, we model the evolution of the enemy strategy from one time step to the next in the Strategy Dynamic Model module, and then we update the strategy probability based on analyst assessment given sensor observation in the *Analyst Reasoning module* and *Strategy Dynamic Module.*

Strategy Transition Model Module: We utilize a Markov transition model to represent the evolution of the enemy strategy, and thus this module models how adversarial strategy transitions when the adversary is confident that it is not being watched. We used a linear map (transition matrix) for starters and will investigate more complicated model in future research. Example of this linear map where the adversary employs three strategies (i.e. s1=Attack; s2=Defend; s3=Deceive), is given below:

$$\Lambda^{k-1} = \begin{bmatrix} 0.95 & 0.25 & 0.25 \\ .025 & .95 & .025 \\ .025 & .025 & .95 \end{bmatrix} : \Sigma^{k-1} \rightarrow \Sigma^{k}$$

The three rows and columns represent the three enemy strategies (attack, defend, deceive). To illustrate, the first row indicates that if the enemy is currently attacking, at the next time stcp, thc cnemy will attack with probability .95, and defend or deceive with probability .025 respectively.

Analyst Reasoning Module: Wc utilizcd a confusion matrix as a function of sensor mode to model the analyst assessment (a more elaborate model can be dcveloped in the future). For instance, GMTI mode has a better ability to detect moving targets and therefore the analyst can better identify the enemy strategy with the observation. The confusion matrices are used to generate random hypothetical observations based on true enemy strategy. The confusion matrices for GMTI and HRR currently utilized are as follows:

$$\Phi^{-1}(\mathrm{GMTI}) = \begin{pmatrix} .7 & .1 & .2 \\ .2 & .4 & .4 \\ .4 & .2 & .4 \end{pmatrix} \quad \Phi^{-1}(\mathrm{HRR}) = \begin{pmatrix} .5 & .2 & .3 \\ .2 & .6 & .2 \\ .4 & .1 & .5 \end{pmatrix}$$

To illustrate, the first row of the GMTI confusion matrix indicates than if the enemy is attacking, an analyst receiving a GMTI report will identify the enemy strategy as attacking with probability .7, defending with probability .1, and deceiving with probability .2.

We utilize the confusion matrix corresponding to the sensor mode chosen by the blue force under the blue force action strategy decision. To determine the correct row of the given confusion matrix, we utilized enemy ground truth via the Enemy Dynamic Model.

Enemy Dynamic Module
This module models by a Markov process how an adversary may adapt strategy when detecting surveillance from the ballistic missile defense system. An example is pictorially depicted below:

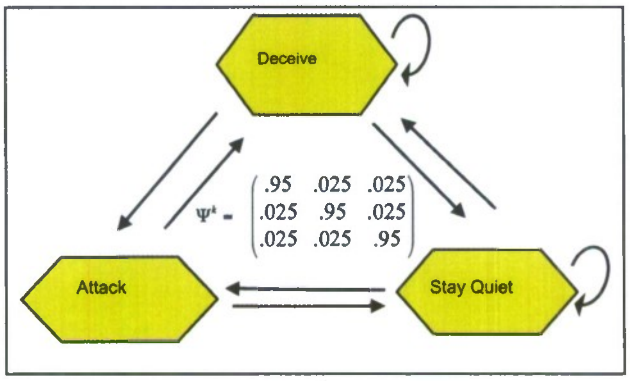

Deceive
Attack
Stay Quiet
$\Psi^t = \begin{pmatrix} .95 & .025 & .025 \\ .025 & .95 & .025 \\ .025 & .025 & .95 \end{pmatrix}$

Figure 14: Markov Chain for Enemy Dynamic Module

2.6.4. Simulation and Results

We conduct a number of Monte Carlo simulations where the initial enemy strategy was selected either randomly or based on the enemy unit composition. In each trial, we conduct 100 time steps where a dynamic sensor decision is made at each step. In the simulation, we try three scenarios where each scenario consists of different composition of units (AA, AB, and BB). We assume that the value of the two unit types can be chosen based on the commander's preference. Similarly, the value of each target types can also be assigned accordingly.

In the simulation, we run 50 Monte Carlo trials for each scenario. In each scenario the window size for sensor action is set to be 10 and the decision threshold is set differently in each test. For example, with threshold equal to 40%, Figures 15-1 thru 15-3 show the results of a typical trial. Figure 15-1 shows that about 36% of the time sensor is off (mode #3) and the rest sensor is on (mode#1 for GMTI, mode#2 for HRR on unit 1, and mode #2 for HRR on unit 2). Figure 15-2 shows that in this trial, the enemy's strategy has changed from 1 (offense) to 2 (defense), then back to 1 and then later move to 3 (deceptive), and back to 1. Figure 15-2 also shows that about 44% of the time, the most likely strategy of our assessment is the true one. Figure 15-3 shows the assessed probability of the three strategies as well as the assessed probability of the true strategy.

In each test, we also compare the game solver with heuristic algorithm where the sensor action/no-action decision was assigned based on a pre-specified probability. We have found out in general, the performance of heuristic solver is significantly worse than the one with gamc solver. This is understandable since the cnemy strategy changes frequently due to our sensor actions. It is much more difficult to assess enemy's strategy without an interactive game theoretic strategy analyzer.

Figures 15-4 thru 15-9 shows the overall performance comparison between the heuristic approach and the game solver approach. In this test, we set the size of time window for enemy strategy policy to be 10 and the decision threshold for change is 80% (i.e., our sensor will need to be on greater than 80% of time in the preceding time window of size 10 before the enemy will potentially change strategy). Note that in the figures, we display both the mean and the one-sigma confidence interval for both probability of correct decision (Pcd) and probability of correct classification (Pcc).

For example, Figure 15-4 compares the average game solver performance with the average heuristic solver performance. The x-axis represents the probability of sensor action for the heuristic approach. Note that in the Figure, the performance is obtaincd by averaging over different cases where each case represents a specific level-1 and level-2 value combination. Figure 15-5, on the other hand, shows the performance of a specific casc (case 13, where the unit type 1 (A) and target type 3 are the emphasis) under the same scenario.

Similarly, Figures 15-6 and 15-7 show the results for scenario BB and Figures 15-8 and 15-9 show the performance of scenario AB. It is clear that in all cases, the game solver outperforms the heuristic solver most of the time. We then change the policy decision threshold from 80% to 101% and test the performance. The corresponding performance results are shown in Figures 15-10 thru 15-15. Note that the fact that the threshold is greater than 100% indicates that the enemy strategy will not change throughout the simulation. In these cases, the heuristic strategy performs almost linearly proportion to the sensor action rate. This is intuitive since no correlation between the enemy strategy and our SRM decision and therefore more sensor action imply more sensor data and better performance. It is interesting to see that the game solver performs approximately the same as the heuristic solver at around 70% sensor action.

We also change the policy decision threshold to 40% and compare the performance. The results are given in Figures 15-16 t0 15-21. The resulting performance curves are somewhat similar to that of Figures 5-9 thru 5-15 except in these cases, the heuristic solver performs better in terms of Pcd when the sensor action rate is small. Also, the overall performance is worse with 40% threshold than that of 101% threshold. Note that with 40% threshold, the enemy strategy changes much more rapidly and therefore is much harder to identify.

Finally, to test the robustness of the game solver, we modify the game solver ratio from 0.5/0.5 to 0.9/0.1. This ratio is to balance between long term and short term prediction/assessment of enemy's strategy. The results are shown in Figures 15-22 thru 15-27. They are very similar to Figure 15-4 thru 15-8 which demonstrates the robustness of the algorithm.

(*sct* below stands for strategy change threshold)

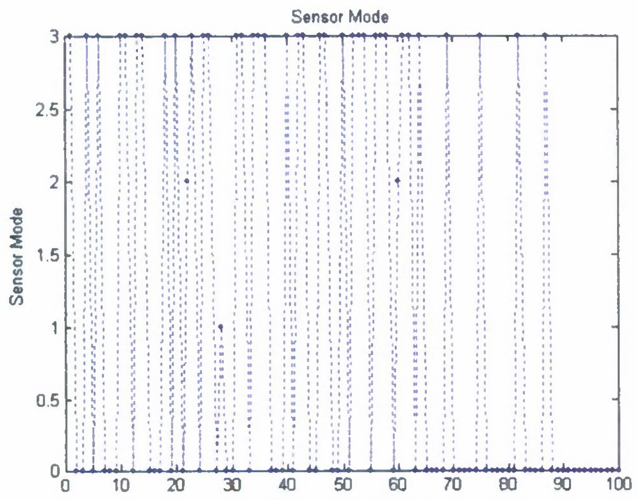


Figure 15-1: Sensor Mode Decision

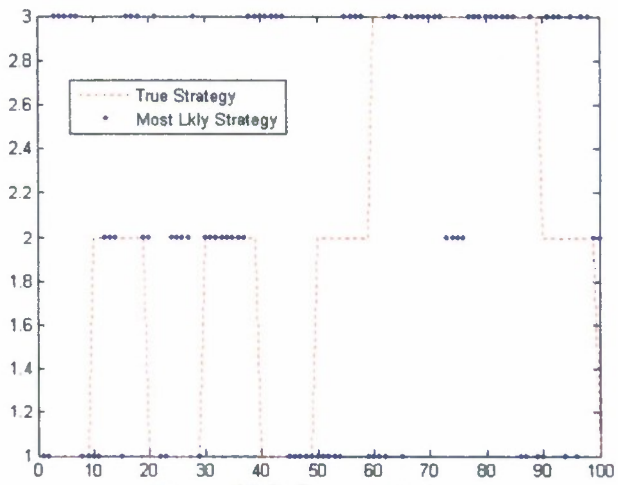


Figure 15-2: Enemy Strategy

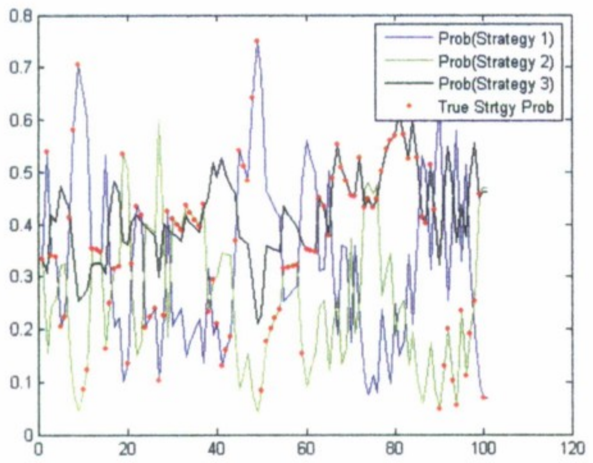


Figure 15-3: Assessed Strategy Probabilities

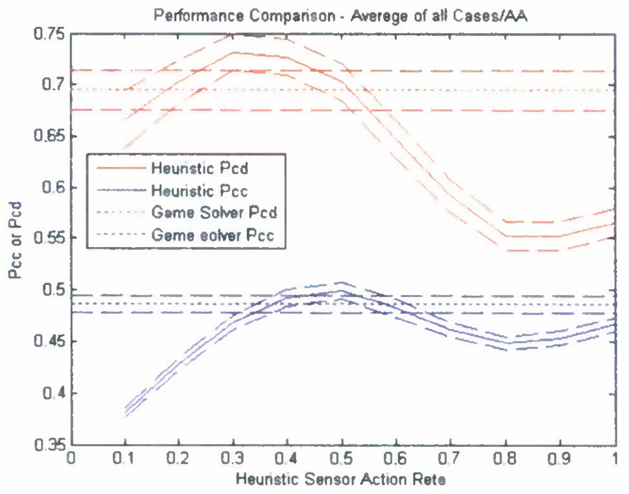


Figure 15-4: AA (sct = .80/ fixed initial strategy (1))

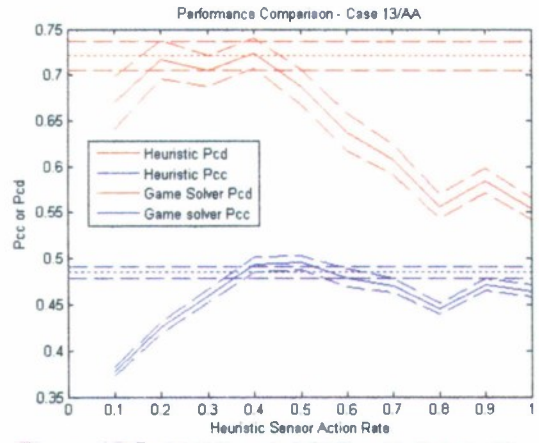


Figure 15-5: AA(13), sct=0.80/fixed initial str. (1)

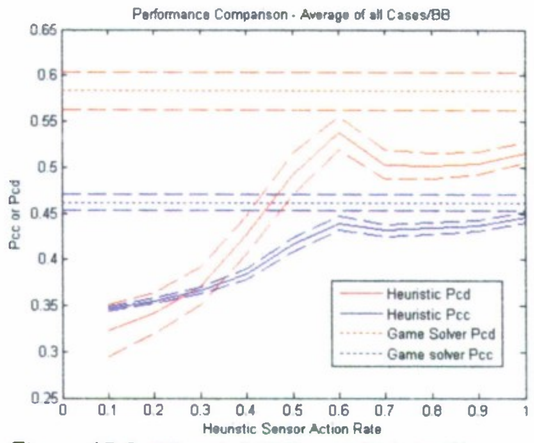


Figure 15-6: BB, sct=0.80/fixed initial str. (2)

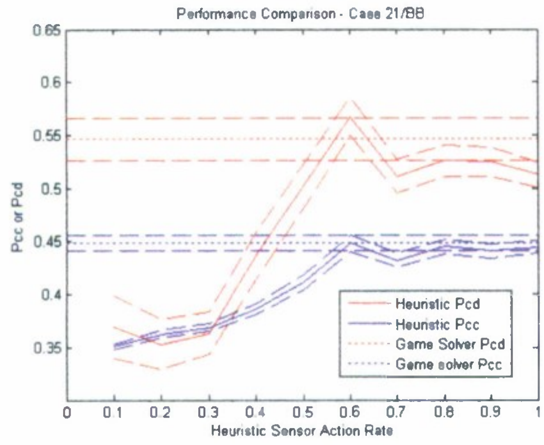


Figure 15-7: Case 21/BB,sct=0.80/fixed initial str.(2)

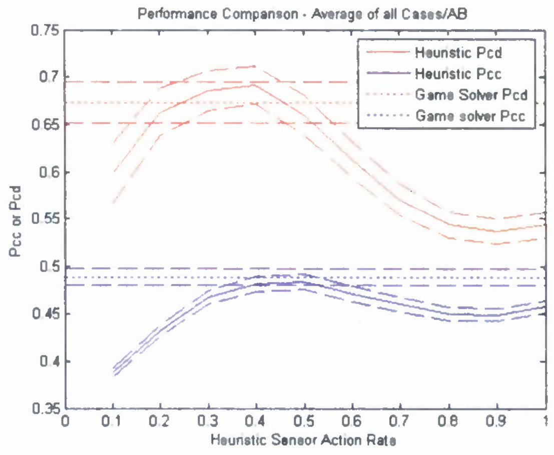


Figure 15-8: AB, sct=.80 and fixed initial str. (2)

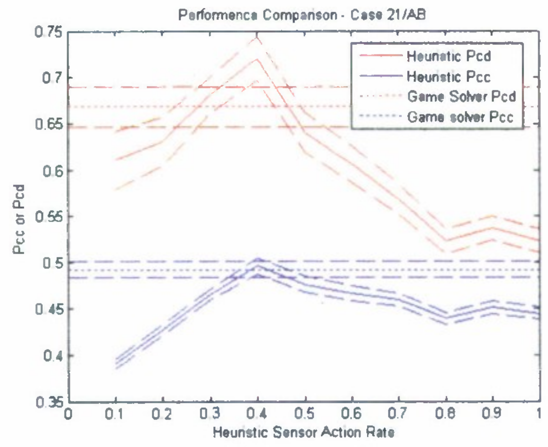


Figure 15-9: Case 21/AB, sct= 0.80/fixed initial str. (2)

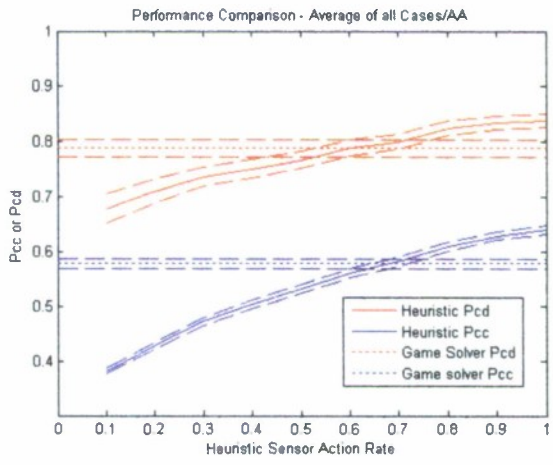


Figure 15-10: AA sct = 1.01/fixed initial str. (1)

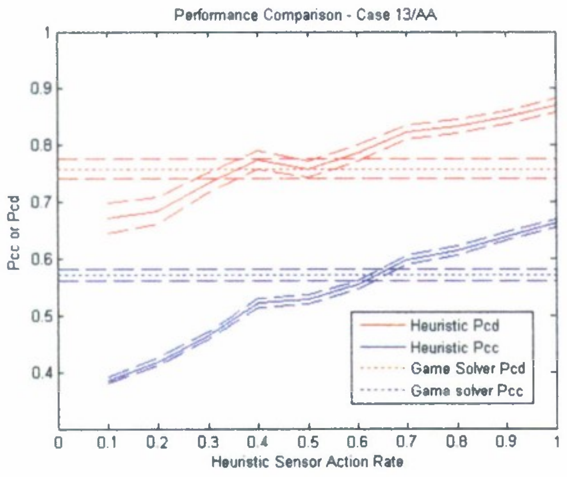


Figure 15-11: Case 13/AA sct=1.01/fixed initial str.(1)

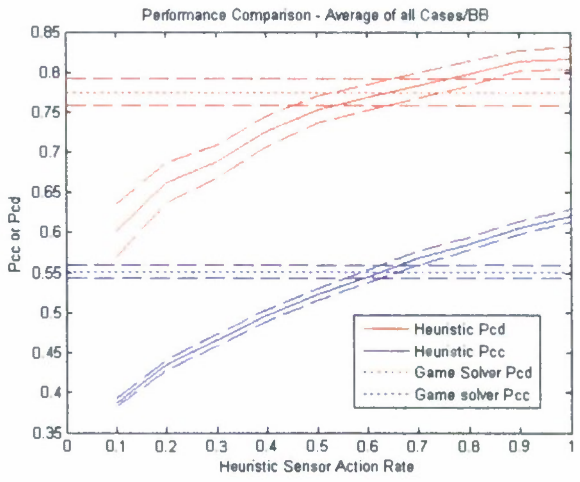


Figure 15-12: BB, sct=1.01/fixed initial str. (2)

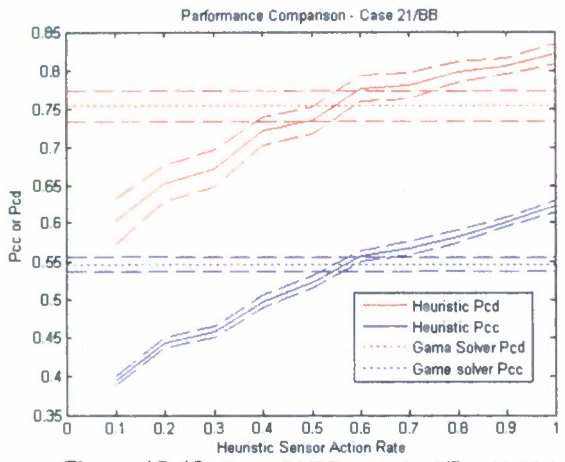


Figure 15-13: Case 21/BB, sct=1.01/fixed initial

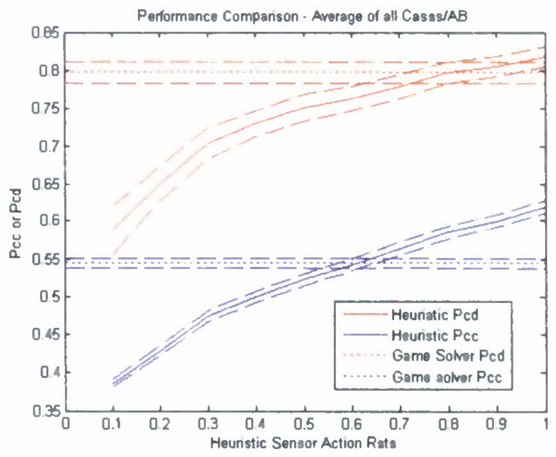


Figure 15-14: AB, sct=: 1.01/fixed initial str. (2)

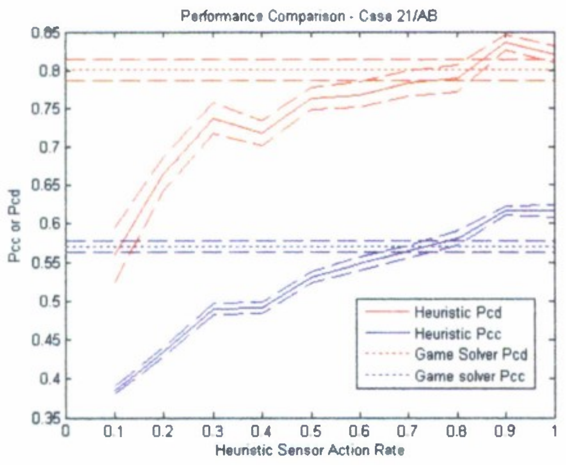


Figure 15-15: Case 21/AB, sct=1.01, fixed initial str.(2)

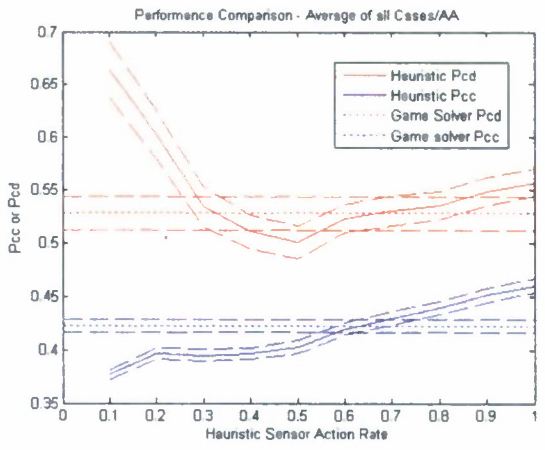


Figure 15-16: AA, sct=0.40/fixed initial str.(1)

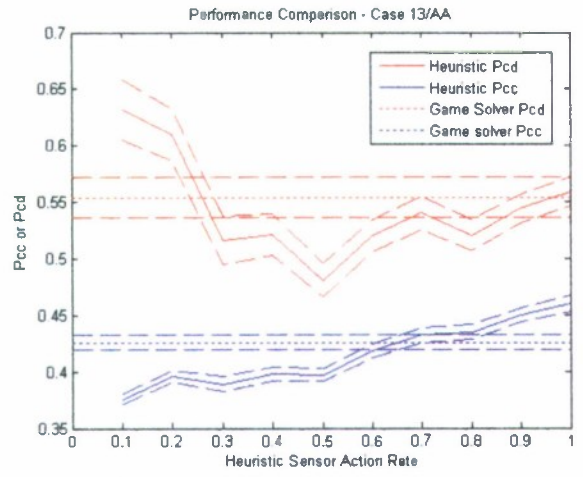


Figure 15-17: Case 13/sct=0.40/fixed initial str(1)

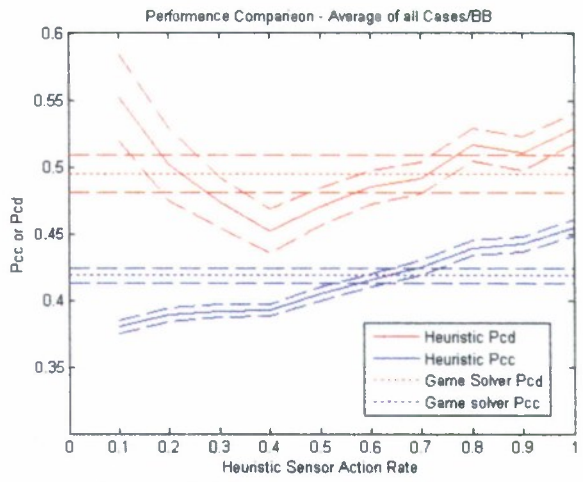


Figure 15-18: BB, sct=0.40/fixed initial str. (2)

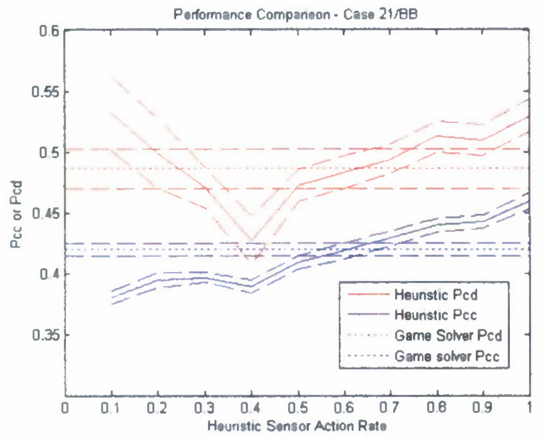


Figure 15-19: Case 21/BB,sct: 0.40/fixed initial str(2)

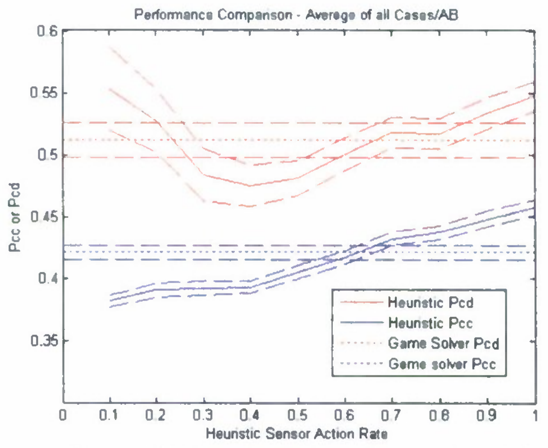


Figure 15-20: sct= 0.40 and fixed initial str.(2)

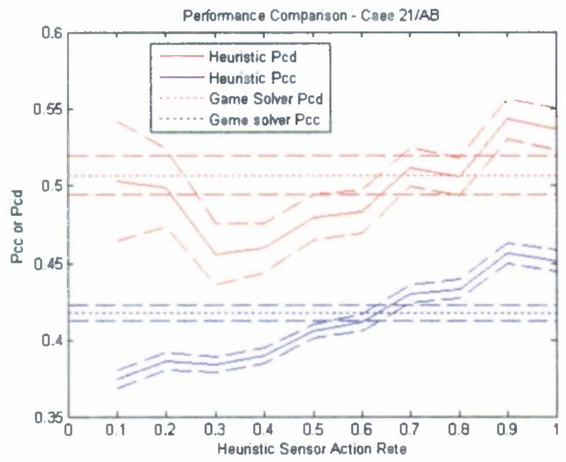


Figure 15-21: Case 21/AB, sct=: 0.40/ fixed initial str.(2)

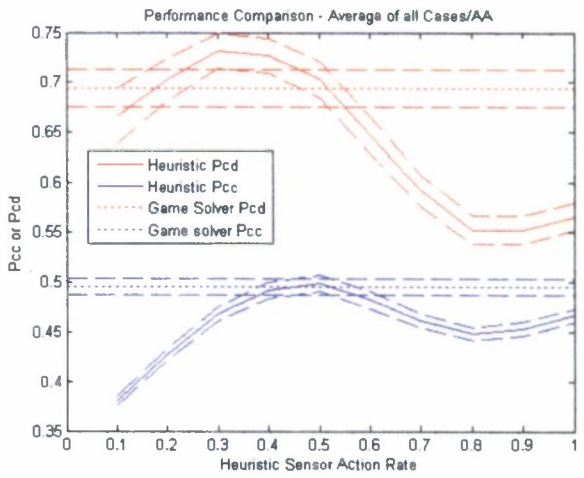


Figure 15-22: AA, game solver ratio 0.9/0.1

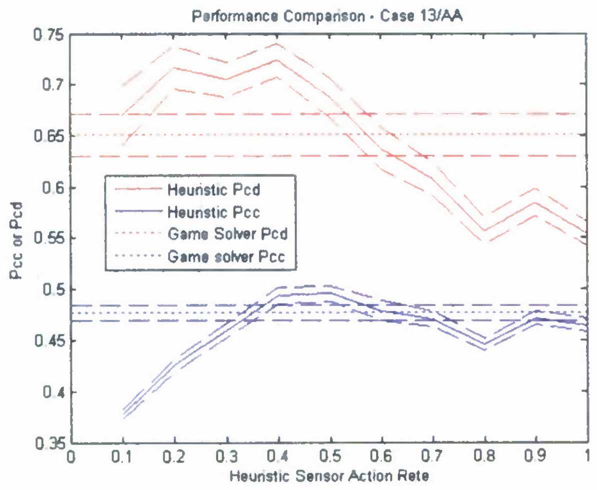


Figure 15-23: Case 13/AA, game solver ratio 0.9/0.1

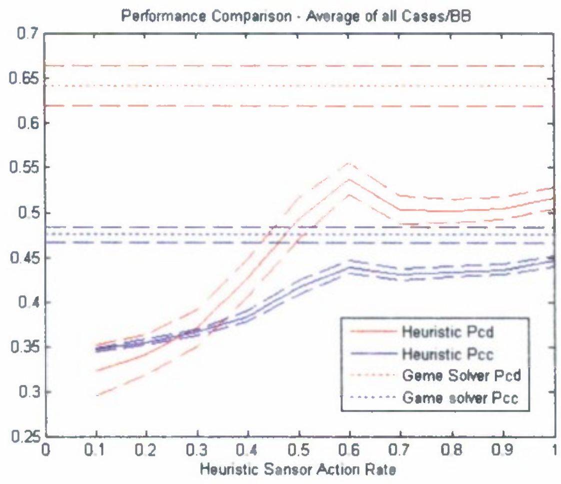


Figure 15-24: BB, game solver ratio 0.9/0.1

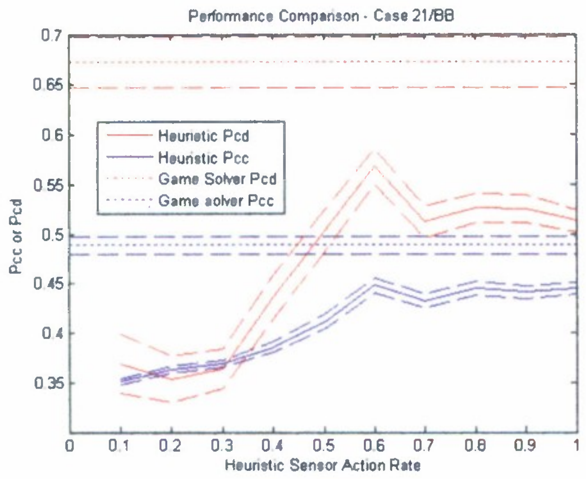


Figure 15-25: Case 21/BB, game solver ratio 0.9/0.1

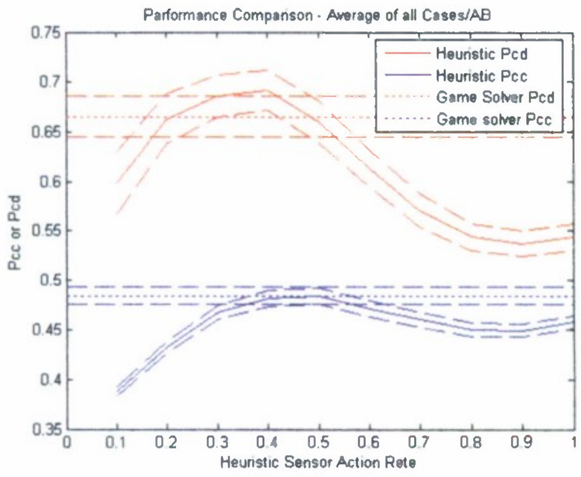


Figure 15-26: AB, game solver ratio 0.9/0.1

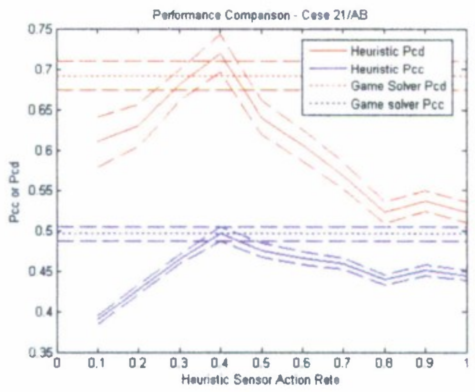


Figure 15-27: Case 21/AB with new game solver ratio 0.9/0.1

2.7. SRM ENVIRONMENT FOR HIGH LEVEL FUSION (GLOBAL)

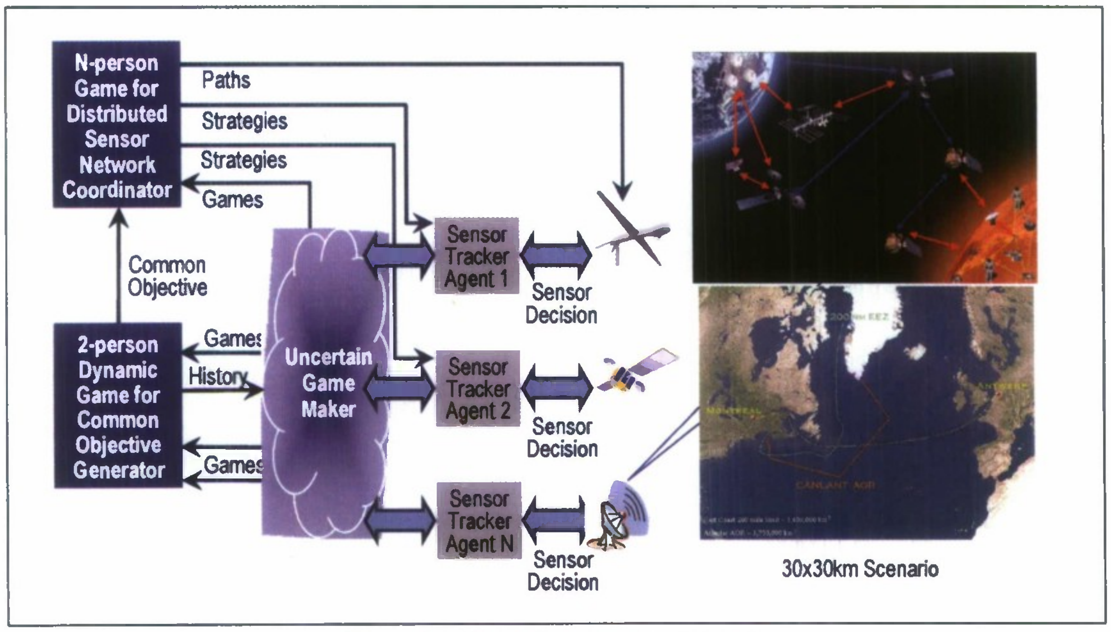


Figure 16: Going from local to Global

Figure 16 depicts how a local game theoretic SRM-Tracker pair (one akin to Figure 14 above) may be put together with other such pairs in a distributive and decentralized architecture. The key distinction between the local picture that was discussed in the previous section (2.6) and the global picture in Figure 16 is the presence of common global objective that was absent and does not arise in the local setting. In other words, as each local SRM-Tracker pair tries its best to maintain its local picture of situational awareness, there exists a global objective that these pairs are supposed to achieve together. Is it always the case that the decisions that Tracker-Sensor Pair has to make to optimize its local objective function coincide with the decisions that it needs to make to contribute maximally to the global objective? This is often the case, but unfortunately, not always true. It turns out that there is *tension* between the local decisions (which are results (i.e. Nash equilibrium) of solving local 2-person game) and the global decisions (which are results (i.e. Nash equilibrium) of solving global N-person game), and each SRM–Tracker pair must be able to choose optimally in order not to neglect one in favor of the other. Exactly how to balance these two competing needs is still very much an area of open research (even in the fields of economics where game theory plays a vital role), and a very difficult problem, which involves solving N-person games, which itself is a difficult problem, as we explained in previous sections. We made some modicum of progress through this research addressing this problem, which we explain now.

2.7.1. Uncertain Global Adversarial Identity

The fact that there is an uncertainty regarding the identity of the global adversary should be of no surprise, as there is an uncertainty in the identity of the local adversaries (which make up the global adversary) as discussed in section 2.6. As such, we tackle this issue in the similar way, as we did for the uncertain games that arose in the local setting. In other words, we will take the probability distribution over all possible global adversary types and use that to define the uncertain game that the global blue force is engaged in against the global red force. To be more precise, since we had the following for the local case (as defined in section 2.6)

- Local adversary types for each TSP = {Offensive, Defensive, Deceptive};

If the number of TSP(Tracker-Sensor-Pair)'s = 3 for the global case, then we have:

- Global adversary types = {Offensive, Defensive, Deceptive}

where

- Global offensive adversary consists of at least two local offensive adversaries
- Global defensive adversary consists of at least two local defensive adversaries
- Global deceptive adversary consists of at least two local deceptive adversaries

Defined as above, number of each global adversarial type is 7 as

- 7 = 1 (3 of a kind) + C(3,2) (2 of a kind) x 2 (remaining 2 choices)

This still leaves cases where each TSP is of a different type. There are 6 (= 3!) of them and they will be classified as deceptive type. Therefore, we have in the end for the global adversary types:

- 7 global offensive adversary consisting of at least two local offensive adversaries
- 7 global defensive adversary consists of at least two local defensive adversaries
- 13 global deceptive adversary consists of at least two local deceptive adversaries or of 3 different local adversary types

And the global adversary type at a given point will be a probability distribution over these three types calculated from the probability distribution of each of the 3 local adversary types.

2.7.2. Solving the global 2-person Game (Common Objective Generator)

As the identity of the global adversary is being determined, so the payoff matrix for the 2-person game that the blue and the red are engaged in is also being determined. We show one particular payoff matrix we use extensively as follows:

$$\text{global\_payoff\_matrix} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -1 & 4 \\ -1 & 4 & -3 \end{bmatrix}$$

As this is the convention, the rows represent the strategy space of the blue (i.e. the sensor network of three TSP's) and the columns represent the strategy space of the global adversary, with the following strategies:

- Global Blue Strategies = {Sense A Lot (SA), Sense Little (SL), Don't Sense (DS)}
- Global Red Strategies = {Attack (AT), Defend (DE), Deceive (DC)}

In other words, global red adversary can employ the overall strategy/objective of "attack", "defend" and "deceive" strategy which it will try to achieve using its local agents to achieve. In response, sensor network can employ the overall strategy/objective of "Sense A Lot" which will then be communicated to the individual sensors in its network, (exactly how this global objective will be translated into a local strategy is a topic of the next section), and the same goes for the global strategy/objective of "Sense Little", and finally for "Don't Sense". Their relative measure of success & failure employing different strategies are reflected in the following table, which coincides with the payoff matrix above:

	Attack (AT)	Defend (DE)	Deceive (DC)
Sense A Lot (SA)	1	2	-1
Sense Medium (SM)	2	-1	4
Sense Little (SL)	-1	4	-3

For example, when the global adversary is in the attack mode, if the sensor system is not employing its distributed sensors to monitor what the adversary is doing (i.e. DS strategy), the sensor system loses its measure of success (-1), and the adversary gains the measure of success (1). To the contrary, if the adversary is in the defensive mode, it is to the sensor network's advantage to not sense (DS), so as to not waste its sensing resources, as the adversary is in the defensive mode, and not at the current time doing something that is worthy of being sensed.

The Nash equilibrium for this pay off matrix is as follows (using the techniques we described about 2-person games):

$$\text{Nash\_blue} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

$$\text{Nash\_red} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

Now in order to estimate the strategy or the intent of the global adversary, we then use the techniques that we developed in the local dynamic 2-person game situation to solve such games as above as in the following linear combination:

$$\begin{aligned}\text{global\_red\_strategy}[t+1] = 0.2 \cdot \text{global\_Nash\_red}[t+1] &+ \\ 0.2 \cdot \text{global\_red\_strategy}[t] &+ \\ 0.2 \cdot \text{local\_red\_strategy\_p1}[t] &+ \\ 0.2 \cdot \text{local\_red\_strategy\_p2}[t] &+ \\ 0.2 \cdot \text{local\_red\_strategy\_p3}[t]&\end{aligned}$$

In other words, in order to estimate the current global strategy, we combinc the global red strategy of the previous time epoch, current *observation* of the adversarial global strategy seen from the current three local sensors and the current global Nash equilibrium (as a baseline estimate).

However, there are a few assumptions that we made in writing down the above, which we must pursue in further research. They are:

1. Even though we have used the equal waiting on all five components that contribute to the current global strategies, other weightings are certainly possible and we will explore more in future research.
2. We have made an assumption that the strategy space of the local adversaries as well as that of the global adversary are all equal to each other, in order to map the local strategy space to global strategy space. Such assumption is rather strong, indeed, and in future research, we will explore exactly what thc inverse (or pseudo-inverse) map Ψ^{-1} of

   $$\Psi : \Lambda^t_{\text{global}} \rightarrow \Lambda^t_{\text{local}}$$

3. We have assume that the transition map:

   $$\Phi^t : \Lambda^t_{\text{local}} \rightarrow \Lambda^{t+1}_{\text{local}}$$

   is a simple identity map. However, this is too simplistic of an assumption and a simple linear map using a transition map, akin to what was done in local case, should be considered (e.g. $\Phi = \begin{bmatrix} 0.9 & 0.1 & 0.1 \\ 0.05 & 0.8 & 0.15 \\ 0 & 0.2 & 0.8 \end{bmatrix}$)

Similarly, for the blue, we proceed as we did before in the local case, i.e., linear combination of Bayesian response, best possible blue response if our estimate of the opponent is a perfect estimate, and Nash response, the equilibrium point, that is considered as a safe response given the opaqueness to the current adversarial strategy, as follows:

$$\begin{aligned} \text{global\_blue\_strategy}[t+1] = {} & 0.5 \cdot \text{blue\_global\_Nash\_strategy}[t+1] + \\ & 0.5 \cdot \text{blue\_global\_Bayesian\_Response}[t+1] \end{aligned}$$

2.7.3. N-person Game for Sensor Network Coordinator

Now once we have a common objective (namely "Sense A Lot (SA)", "Sense Little (SL)", or "Don't Sense (DS)") come out from the Common Objective Generator, as we just described in the previous section, we then have to push down this global common objective to the local levcl and have it be distributed among TSP's (Tracker-Sensor Pair). Such push-down and distribution of the global objective can happen in two ways:

1. If the communication among the TSP's are robust), it can happen cooperatively among the TSP's and one way that this can be implemented is via the proccss of finding endoge-

nous equilibrium thru cooperative fictitious game structure of Patek, et. al., as was outlined section 2.5

2. If the communication among TSP's is limited or compromised, each of the common objective can then be translated into a non-cooperative game which then each TSP can then play against other TSP's to ensure the global objective is met via their competition, rather than cooperation.

In the current research, we chose the method #2 via Uncertain Game Maker, though # 1 is certainly worthy of further investigation and future research.

2.7.4. Uncertain Game Maker

We first let

$$i = \arg\max_{1 \le j \le 3}(\text{blue\_global\_strategy}(j))$$

Now if $i = 1$ (i.e. global objective = "Sense A Lot (SA)"), we have the following 3-person matrix game: (recall, as is convention, rows represent strategy of first player (P1), and columns represent strategies of the second player (P2). Also recall that locally each sensing agent has two strategies: {Sense} and {Not Sense}):

P3 = {Sense}	Sense	Not Sense
Sense	3	2
Not Sense	2	1

P3 = {Not Sense}	Sense	Not Sense
Sense	2	1
Not Sense	1	0

Nash solution for this matrix game, using the principle of dominance is (1, 1, 1) as the pure Nash equilibrium strategy.

Now, if $i = 2$ (i.e. the global objectivc = "Sense Medium (SM)"), we then have:

P3 = {Sense}	Sense	Not Sense
Sense	3	2
Not Sense	2	3

P3 = {Not Sense}	Sense	Not Sense
Sense	2	3
Not Sense	3	0

In this case, there is no pure Nash solution for this payoff matrix, which makes things more interesting. Also, notice there is payoff of 3 for (1, 1, 1). This reflects a bias towards coordination to achieve the common objective of "Sensing A Little (SM)".

Finally, if *i = 3* (i.e. the global objective = "Sense Little (SL)"), we then have:

P3 = {Sense}	Sense	Not Sense
Sense	1	1
Not Sense	1	2

P3 = {Not Sense}	Sense	Not Sense
Sense	1	2
Not Sense	2	1

Nash solutions for this matrix game are {(2, 2, 1), (2,1,2), (1, 2, 2)} as pure Nash equilibriums and

$\{\frac{1}{2}(2,1,2)+\frac{1}{2}(1,2,2)\}$ as a mixed Nash solution.

Now whichever the global objective may be (thus whichever game we may be solving), the local strategy supporting the global objective will then be pushed down to each local sensing agent (TSP) and it is a (new) job/capability of each TSP to decide what to do with this new globally-inspired objective.

2.7.5. Balancing Global and Local Needs

The figure below (Figure 17) captures pictorially the decision that each TSP needs to make. Locally, at a given time, using the local L3 module that was described in section 2.6, each TSP needs to decide whether to activate its sensor or not (Level 3 decision) which then employs its Level 1 / Level 2 SRM algorithms to determine whether to use MTI mode or HRR mode. Furthermore, it also receives from Global Sensor Network Coordinator, as described in sections 2.7.4 and 2.7.5, a decision whether to sense or not to sense, so as to help achieve the global objective in concert with other TSP's.

Currently, for starters, we are using the equal balance between the two. In other words, we havc

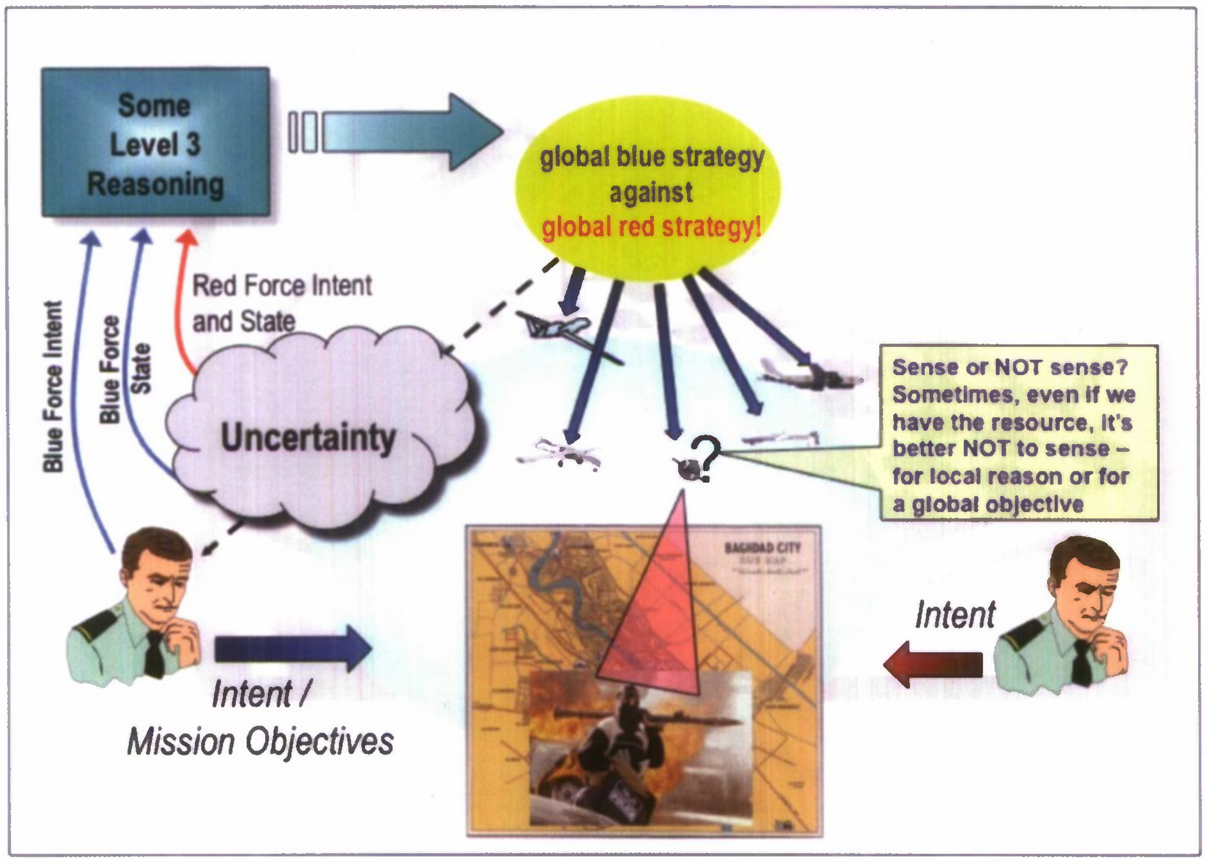

Some Level 3 Reasoning
global blue strategy against global red strategy!
Red Force Intent and State
Blue Force Intent
Blue Force State
Uncertainty
Sense or NOT sense? Sometimes, even if we have the resource, it's better NOT to sense – for local reason or for a global objective
BAGHDAD CITY
Intent
Intent / Mission Objectives

Figure 17: Balancing between local and global

$$s_{TSP} = \frac{1}{2} s_{local} + \frac{1}{2} s_{global}$$

In future, a more effective balancing algorithm will bc employed. The one currently envisioned takes into account the size of uncertainty that grows or shrinks over time at the global levcl and also at the local level. In other words, we could have:

$$s_{TSP} = \frac{k_1}{size(\text{local uncertainty})} s_{local} + \frac{k_2}{size(\text{global uncertainty})} s_{global}$$

2.7.6. Putting It Altogether

We now put all these components together and run this hierarchical simulation loop as described in the following pseudo-code:

```
<hierarchical distributed game theoretic simulation algorithm>
==================================================================
for (each simulation step)

% First play three separate local 2-person games
        for (each player)
                Dynamic 2-person game_solver_for local sensing strategy
                If (sense)
                        activate_sensor using SRM Module
                        Evolve enemy strategy using Strategy Transition Module
                        Monitor enemy strategy using Analyst Reasoning Module
                        Update enemy strategy using Enemy Dynamics Module
                end
        end

% Now play global 2-person game
        Calculate next global strategy via CommonObjectiveGenerator using
                Previous global strategy and previous local strategies.
        Find next global n-person game via UncertainGameMaker using
                current global strategy and local adversary types
        Find next globally inspired local strategies by solving the current n-person
                game via n_person_game_solver

% Final Balancing act
        Compute new sensing strategies by balancing local sensor strategies
                and globally inspired local sensing strategies.

end;
==================================================================
```

And these algorithms are pictorially captured in the Figure 18 below:

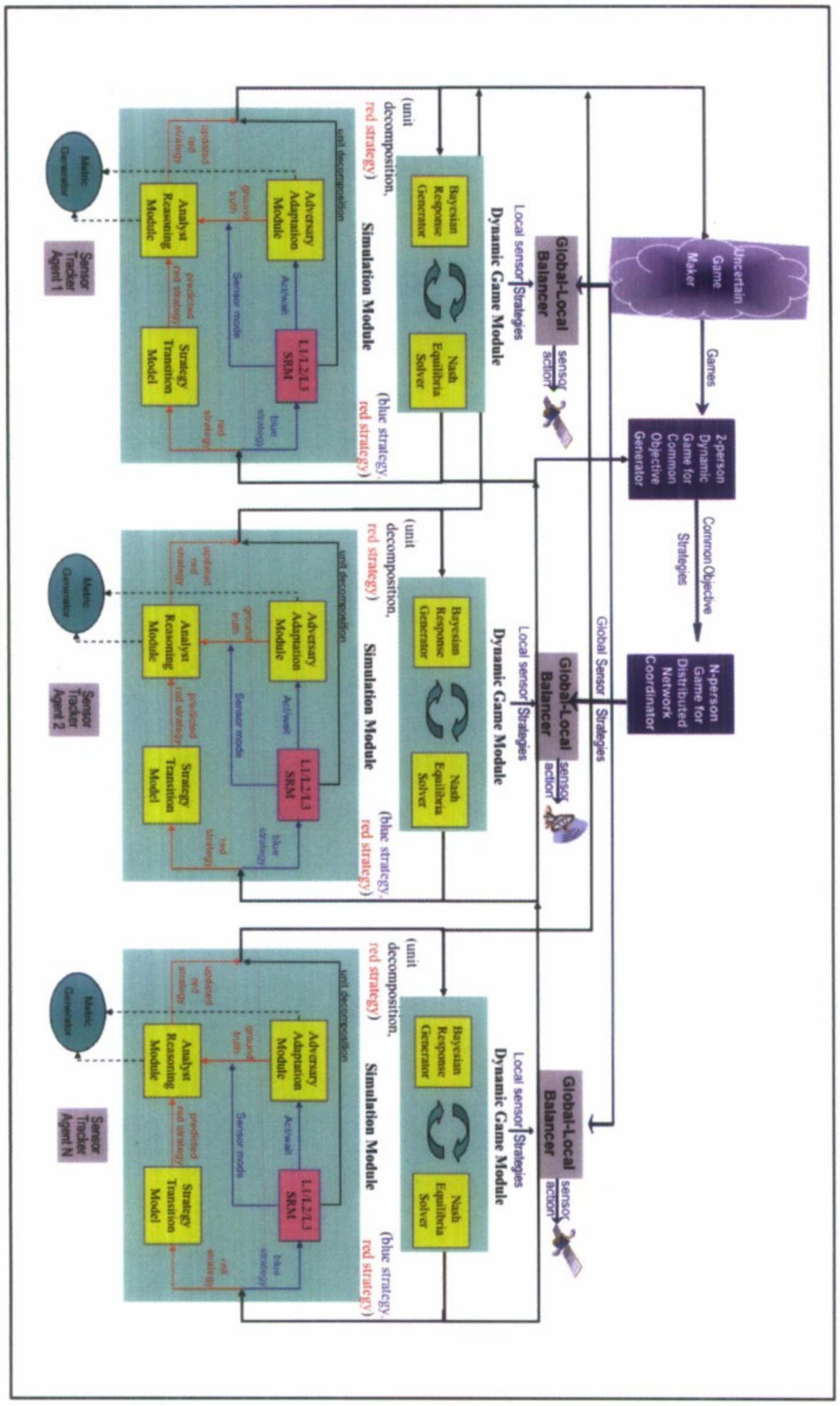


Figure 18: Final Simulation Architecture

3. REFERENCES

[1] John A. Tirpak, "Space and Counterpace," in *Journal of the Air Force Association,* vol.89, no. 6, June 2006.
[2] A. Haurie, J.B. Krawcczyk, "An Introduction to Dynamic Games," in *Journal of the Air Force Association,* pg. 79-82.
[3] Rufus Isaacs, "A Mathematical Theory with Applications to Warfare and Pursuit, Control, and Optimization" *Dover Publications.*
[4] Sang Chin, Scott Laprise, KC Chang, "Game-theoretic Model-based Approach to Higher-Level Fusion with Application to Sensor Resource Management" in *Proceedings of National Symposium on Sensor and Data Fusion, 2006*
[5] John F. Nash, "Non-cooperative Games" in *Ph.D Thesis, Princeton University, 1951*
[6] A. Garcia, S. Patek and K. Sinha, "A Decentralized Approach for Simulation-based Optimization", Operations Research (2007) Vol. 55 (4) pp 1-16
[7] Y. Zhao, P. A. Beling, and S. D. Patek, " Coordination Issues in Multi-Agent Bayesian Search Problems ", submitted for publication